日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 Date of Application:	２００４年　６月　１日
出願番号 Application Number:	特願２００４－１６３６４９
パリ条約による外国への出願に用いる優先権の主張の基礎となる出願の国コードと出願番号 The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is	ＪＰ２００４－１６３６４９
出願人 Applicant(s):	株式会社ソニー・コンピュータエンタテインメント

２００５年　５月１１日

特許庁長官
Commissioner,
Japan Patent Office

小川

【書類名】　特許願
【整理番号】　SCEI03033
【提出日】　平成16年 6月 1日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 9/00
G06F 15/00
【発明者】
【住所又は居所】　東京都港区南青山２丁目６番２１号　株式会社ソニー・コンピュータエンタテインメント内
【氏名】　安達　健一
【発明者】
【住所又は居所】　東京都港区南青山２丁目６番２１号　株式会社ソニー・コンピュータエンタテインメント内
【氏名】　矢澤　和明
【発明者】
【住所又は居所】　東京都港区南青山２丁目６番２１号　株式会社ソニー・コンピュータエンタテインメント内
【氏名】　瀧口　巌
【発明者】
【住所又は居所】　東京都港区南青山２丁目６番２１号　株式会社ソニー・コンピュータエンタテインメント内
【氏名】　今井　敦彦
【発明者】
【住所又は居所】　東京都港区南青山２丁目６番２１号　株式会社ソニー・コンピュータエンタテインメント内
【氏名】　田村　哲司
【特許出願人】
【識別番号】　395015319
【氏名又は名称】　株式会社ソニー・コンピュータエンタテインメント
【代理人】
【識別番号】　100105924
【弁理士】
【氏名又は名称】　森下　賢樹
【電話番号】　03-3461-3687
【手数料の表示】
【予納台帳番号】　091329
【納付金額】　16,000円
【提出物件の目録】
【物件名】　特許請求の範囲　1
【物件名】　明細書　1
【物件名】　図面　1
【物件名】　要約書　1

【書類名】特許請求の範囲
【請求項１】
プロセッサによってタスクを実行する際、処理の単位時間を、リアルタイム性を保証するための保護帯域とリアルタイム性を保証しない非保護帯域に分割し、プロセッサの処理能力が低下したとき、非保護帯域において実行すべきタスクの実行を適宜スキップすることを特徴とするタスク管理方法。
【請求項２】
請求項１に記載の方法において、プロセッサまたはその周辺回路の温度が所定のしきい値を越えたときプロセッサの動作周波数が低減されることを特徴とするタスク管理方法。
【請求項３】
請求項１に記載の方法において、プロセッサの消費電力に応じてプロセッサの動作周波数が低減されることを特徴とするタスク管理方法。
【請求項４】
プロセッサによって実行すべきタスクをその性質に応じて第１タイプと第２タイプへ分類し、所定の要因で処理のリアルタイム性が損なわれる可能性があるとき、第１タイプのタスクを実行しつつ、第１タイプのタスクの合間に実行されるべき第２タイプのタスクの実行を適宜スキップすることを特徴とするタスク管理方法。
【請求項５】
請求項４に記載の方法において、前記第１タイプのタスクはリアルタイム性を保証すべきタスクとして所定の方法でプロセッサが認識するものであり、前記第２タイプのタスクはリアルタイム性を保証しないタスクとして所定の方法でプロセッサが認識するものであることを特徴とするタスク管理方法。
【請求項６】
主処理部に実行させる複数のタスクの切り替え指示を行う切替指示部と、
前記主処理部の処理能力を検出する検出部と、
を備え、
前記切替指示部は、処理の単位時間を、リアルタイム性を保証するための保護帯域とリアルタイム性を保証しない非保護帯域に分割し、前記主処理部の処理能力が低下したとき、非保護帯域において実行すべきタスクの実行を適宜スキップさせることを特徴とするタスク管理装置。
【請求項７】
請求項６に記載の装置において、前記検出部は、前記主処理部における動作周波数を検出することを特徴とするタスク管理装置。
【請求項８】
請求項６または７に記載の装置において、各タスクにて実行されるプログラム内に記述されたリアルタイム性に対する要請を解釈する解釈部を更に備え、
前記切替指示部は、その解釈に基づいて各タスクを前記保護帯域または前記非保護帯域のいずれかに配することを特徴とするタスク管理装置。
【請求項９】
請求項６または７に記載の装置において、各タスクにて実行されるプログラムの性質を判断する判断部を更に備え、
前記切替指示部は、その判断に基づいて各タスクを前記保護帯域または前記非保護帯域のいずれかに配することを特徴とするタスク管理装置。
【請求項１０】
請求項６から９のいずれかに記載の装置において、前記単位時間は表示に関連する単位時間であることを特徴とするタスク管理装置。
【請求項１１】
請求項６から１０のいずれかに記載の装置において、前記主処理部の使用率を検出する検出部を更に備え、
前記切替指示部は、前記使用率に応じて非保護帯域において実行すべきタスクの実行率

を変更することを特徴とするタスク管理装置。
【請求項１２】
請求項１１に記載の装置において、前記主処理部の処理能力に関する情報と、その処理能力での非保護帯域において実行すべきタスクの実行率とを対応づけて保持するテーブルを更に備え、前記切替指示部は、前記主処理部の使用率が所定のしきい値より低い場合、非保護帯域において実行すべきタスクの実行率を、前記テーブルで設定された実行率より上げることを特徴とするタスク管理装置。
【請求項１３】
主処理部に実行させる複数のタスクの切り替え指示を行う切替指示部と、
前記主処理部の処理能力を検出する検出部と、
を備え、
前記切替指示部は、前記主処理部に実行されるべきタスクをその性質に応じて第１タイプと第２タイプへ分類し、所定の要因で処理のリアルタイム性が損なわれる可能性があるとき、第１タイプのタスクを実行しつつ、第１タイプのタスクの合間に実行されるべき第２タイプのタスクの実行を適宜スキップすることを特徴とするタスク管理装置。
【請求項１４】
所定のタスクを実行する主処理部と、
処理の単位時間を、リアルタイム性を保証するための保護帯域とリアルタイム性を保証しない非保護帯域に分割し、前記主処理部の処理能力が低下したとき、非保護帯域において実行すべきタスクの実行を適宜スキップさせるタスク管理部と、
を備えることを特徴とする半導体集積回路。
【請求項１５】
請求項１４に記載の回路において、前記主処理部に所定の動作周波数のクロックを供給するクロック生成部を更に備え、
前記タスク管理部は、前記動作周波数が低下したとき、非保護帯域において実行すべきタスクの実行を適宜スキップさせることを特徴とする半導体集積回路。
【請求項１６】
請求項１５に記載の回路において、前記主処理部またはその周辺の温度が所定のしきい値を越えたとき、前記クロック生成部は前記動作周波数を低減することを特徴とする半導体集積回路。
【請求項１７】
請求項１６に記載の回路において、消費電力に応じて、前記クロック生成部は前記動作周波数を低減することを特徴とする半導体集積回路。
【請求項１８】
請求項１５に記載の回路において、前記タスク管理部は、前記主処理部またはその周辺の温度が所定のしきい値を越えたとき、非保護帯域において実行すべきタスクの実行を適宜スキップさせることを特徴とする半導体集積回路。
【請求項１９】
請求項１５に記載の回路において、前記タスク管理部は、消費電力に応じて非保護帯域において実行すべきタスクの実行を適宜スキップさせることを特徴とする半導体集積回路。
【請求項２０】
所定の動作周波数でタスクを実行する主処理部と、
前記動作周波数のクロックを前記主処理部に供給するクロック生成部と、
タスク管理機能を実現するためのプログラムを外部から読み込むことによって動的にタスク管理機能を実現するための回路と、
を備え、
前記タスク管理機能は、処理の単位時間を、リアルタイム性を保証するための保護帯域とリアルタイム性を保証しない非保護帯域に分割し、前記動作周波数が低下したとき、非保護帯域において実行すべきタスクの実行を適宜スキップさせることを特徴とする半導体

累積回路。
【請求項２１】
所定の動作周波数でタスクを実行するプロセッサと、
前記プロセッサに実行させるプログラムを保持する記憶部と、
を備え、
前記プログラムは、処理の単位時間を、リアルタイム性を保証するための保護帯域とリアルタイム性を保証しない非保護帯域に分割し、前記動作周波数が低下したとき、非保護帯域において実行すべきタスクの実行を適宜スキップするようタスクをスケジューリングする機能を前記プロセッサに実現させることを特徴とする電子装置。
【請求項２２】
請求項２１に記載の装置において、前記プロセッサまたはその周辺回路の温度が所定のしきい値を越えたとき前記動作周波数を低減する周波数制御部を更に備えることを特徴とする電子装置。
【請求項２３】
請求項２１に記載の装置において、消費電力に応じて前記動作周波数を低減する周波数制御部を更に備えることを特徴とする電子装置。
【請求項２４】
プロセッサによってタスクを実行する際、処理の単位時間を、リアルタイム性を保証するための保護帯域とリアルタイム性を保証しない非保護帯域に分割し、プロセッサの処理能力が低下したとき、非保護帯域において実行すべきタスクの実行を適宜スキップする機能をコンピュータに実現させるためのプログラム。
【請求項２５】
プロセッサによって実行すべきタスクをその性質に応じて第１タイプと第２タイプへ分類し、所定の要因で処理のリアルタイム性が損なわれる可能性があるとき、第１タイプのタスクを実行しつつ、第１タイプのタスクの合間に実行されるべき第２タイプのタスクの実行を適宜スキップする機能をコンピュータに実現させるためのプログラム。
【請求項２６】
所定の動作周波数でタスクを実行するプロセッサと、
前記動作周波数のクロックを前記プロセッサに供給するクロック生成部と、
前記プロセッサに実行させる複数のタスクの切り替え指示を行う切替指示部と、
を備え、
前記切替指示部は、処理の単位時間を、リアルタイム性を保証するための保護帯域とリアルタイム性を保証しない非保護帯域に分割し、前記プロセッサの動作周波数が低下したとき、非保護帯域において実行すべきタスクの実行を適宜スキップさせることを特徴とするタスク管理システム。
【請求項２７】
請求項２６に記載のシステムにおいて、前記クロック生成部は、前記プロセッサまたはその周辺回路の温度が所定のしきい値を越えたとき前記動作周波数を低減することを特徴とするタスク管理システム。
【請求項２８】
請求項２６に記載のシステムにおいて、前記クロック生成部は、前記プロセッサの消費電力に応じて前記動作周波数を低減することを特徴とするタスク管理システム。

【書類名】明細書
【発明の名称】タスク管理方法、タスク管理装置、半導体集積回路、電子装置、およびタスク管理システム
【技術分野】
【０００１】
この発明は、プロセッサに実行されるタスクを管理する方法、その方法を利用してタスクを管理する装置、その装置の実体としての半導体集積回路、およびその半導体集積回路を備える電子装置に関する。
【背景技術】
【０００２】
ＬＳＩ設計において製造プロセスの微細化と素子の高集積化が一段と進み、チップの性能限界として発熱量を考慮することが設計上非常に重要になってきている。チップが高温になると、動作不良を起こしたり、長期信頼性が低下したりするため、様々な発熱対策がとられている。たとえば、チップの上部に放熱フィンを設けて、チップから発生する熱を逃がす方法がとられる。
【０００３】
また、チップの消費電力分布にもとづいて、プロセッサのタスクをスケジューリングすることも検討されている。また、チップが高温になることを回避する方法として、プロセッサにおける動作周波数を低くすることも検討されている。（たとえば、特許文献１参照）。
【特許文献１】米国特許出願公開第２００２／００６５０４９号明細書
【発明の開示】
【発明が解決しようとする課題】
【０００４】
動作周波数を低くすることにより、チップの発熱量を下げることはできるが、単位時間における処理ステップ数が減ってしまうため、単位時間内に完了すべきタスクが単位時間内に完了できない場合がある。このため、予め最低の動作周波数ベースでプログラムの設計を行う必要がある。しかしこれでは、プロセッサの処理能力が十分に発揮されない。
【０００５】
本発明はこうした課題に鑑みてなされたものであり、その目的は、こうした二律背反する目標を実現し、効果的なタスクの管理方法、その方法に基づくタスク管理装置、その装置の実体としての半導体集積回路、およびその半導体集積回路を備える電子装置を提供することにある。
【課題を解決するための手段】
【０００６】
本発明のある態様は、タスクの実行方法に関する。この方法は、プロセッサによってタスクを実行する際、処理の単位時間を、リアルタイム性を保証するための保護帯域とリアルタイム性を保証しない非保護帯域に分割し、プロセッサの処理能力が低下したとき、非保護帯域において実行すべきタスクの実行を適宜スキップする。厳密には、タスクを実行する、またはタスクをスキップするというのは、そのタスクに関するプロセス、すなわちプログラムを実行またはスキップすることだが、以下簡潔のため単に「タスクを実行する」または「タスクをスキップする」などのように表現する。「タスク」はプログラム上のボリュームに関係なく、ひとつの機能のまとまりを指すとする。したがって、実際に普及しているＯＳが把握するタスクよりも大きな機能単位を指すこともある。
【０００７】
「プロセッサの処理能力の低下」は、プロセッサの動作周波数の低減に起因することであってよい。プロセッサの動作周波数は、プロセッサまたはその周辺回路の温度が所定のしきい値を越えたときに低減されてよい。また、プロセッサの消費電力に応じて低減されてもよい。
【０００８】

この態様によれば、プロセッサの処理能力が低下したとき、非保護帯域において実行すべきタスクの実行が適宜スキップされるので、保護帯域において実行されるべきタスクのリアルタイム性が保証されるか、少なくともその確度が大幅に増す（ただし、以降単に「保証される」という）。言い換えれば、リアルタイム性を保証すべきタスクを、保護帯域に含まれるように設計しておけば、保護帯域を割り込まない範囲でプロセッサの処理能力を変更させることができ、プロセッサの熱制御を柔軟に行うことができる。

【０００９】

本発明の別の態様はタスク管理装置である。この装置は、プロセッサに実行させる複数のタスクの切り替え指示を行う切替指示部と、プロセッサの処理能力を検出する検出部とを備え、切替指示部は、処理の単位時間を、リアルタイム性を保証するための保護帯域とリアルタイム性を保証しない非保護帯域に分割し、プロセッサの処理能力が低下したとき、非保護帯域において実行すべきタスクの実行を適宜スキップさせる。

【００１０】

この装置は、各タスクにて実行されるプログラム内に記述されたリアルタイム性に対する要請を解釈する解釈部を更に備えてもよく。その場合、切替指示部は、その解釈に基づいて各タスクを保護帯域または非保護帯域のいずれかに配してもよい。「リアルタイム性に対する要請」は、例えばタスクを保護帯域で実行すべきか否かの判定に利用可能な情報であってよく、プログラム内に記述された属性を示す情報であってもよいし、タスクの重要度や優先度を示す情報であってもよい。

【００１１】

この装置は、各タスクにて実行されるプログラムの性質をプロセッサで判断する判断部を更に備えてもよく。その場合、切替指示部は、その判断に基づいて各タスクを保護帯域または非保護帯域のいずれかに配してもよい。「プログラムの性質」は、そのプログラムで呼び出される命令でもよいし、そのプログラムによるプロセッサの占有率や占有時間であってもよく、タスクを実行することにより間接的に得られる特徴であってよい。

【００１２】

本発明の別の態様はタスク管理システムである。このシステムは、所定の動作周波数でタスクを実行するプロセッサと、その動作周波数のクロックをプロセッサに供給するクロック生成部と、プロセッサに実行させる複数のタスクの切り替え指示を行う切替指示部とを備え、切替指示部は、処理の単位時間を、リアルタイム性を保証するための保護帯域とリアルタイム性を保証しない非保護帯域に分割し、プロセッサの動作周波数が低下したとき、非保護帯域において実行すべきタスクの実行を適宜スキップさせる。

【００１３】

なお、以上の構成要素の任意の組合せ、本発明の表現を方法、装置、システム、コンピュータプログラムなどの間で変換したものもまた、本発明の態様として有効である。

【発明の効果】

【００１４】

本発明によれば、リアルタイムタスクと非リアルタイムタスクとをマルチタスクで実行している際にプロセッサの処理能力が低下した場合でも、リアルタイムタスクのリアルタイム性を保証できる。

【発明を実施するための最良の形態】

【００１５】

実施の形態を説明する前に課題を明確にする。例えばゲームの場合、フレーム期間にＣＧのレンダリングなどの描画処理が完了するように設計される。このように所定の時間内で処理が完了しなければいけないタスクを、以下「リアルタイムタスク」とよぶ。リアルタイムタスクのように時間制限が設けられていないタスクを、以下「非リアルタイムタスク」とよぶ。こうした時間に対する設計思想が異なる２種類のタスクを並行して実行する場合、リアルタイムタスクのリアルタイム性は、非リアルタイムタスクの処理時間に大きく影響される。

【００１６】

図１（ａ）は、通常時の動作周波数ｆ０で動作するプロセッサにおけるタスクの処理状態を時系列的に示す図である。時刻Ｔ０からＴ１、時刻Ｔ１からＴ２、および時刻Ｔ２からＴ３は、それぞれ１フレーム期間である。本図では、フレームの描画を行うためのリアルタイムタスクＲＴ、および２つの非リアルタイムタスクＮＲＴが順番に実行されている。時刻Ｔ１、Ｔ２、Ｔ３からフレームの表示が行われるとする。時刻Ｔ０からＴ１までの間に、リアルタイムタスクＲＴ、第１非リアルタイムタスクＮＲＴ１、および第２非リアルタイムタスクＮＲＴ２が順次実行される。その後、時刻Ｔ２で表示するフレームを描画するリアルタイムタスクＲＴが実行され、続いて第１非リアルタイムタスクＮＲＴ１、第２非リアルタイムタスクＮＲＴ２が順次実行される。時刻Ｔ１、Ｔ２、Ｔ３にフレームを表示するためのリアルタイムタスクＲＴは、それぞれ時刻Ｔ１、Ｔ２、Ｔ３より前に処理が完了しているため、コマ落ちすることなくフレームが表示される。

【００１７】

図１（ｂ）は、動作周波数０．８f_0の場合のプロセッサにおけるタスクの処理状態を時系列的に示す図である。プロセッサの動作周波数は、例えば、プロセッサにおける熱制御のために調整され、プロセッサの温度が高くなると動作周波数が低く調整される。動作周波数が低くなると、フレーム期間におけるプロセッサの実行サイクル数が少なくなる。このため、各タスクを完了するまでの時間が長くなり、フレーム期間毎に実行されるべきリアルタイムタスクＲＴ、第１非リアルタイムタスクＮＲＴ１、および第２非リアルタイムタスクＮＲＴ２が、本来実行されるべきフレーム期間を越えて次のフレーム期間で実行されるようになる。本図では、リアルタイムタスクＲＴのリアルタイム性は、ほぼ保たれるものの、長期的にはコマ落ちが生じてしまう。

【００１８】

図１（ｃ）は、動作周波数０．７f_0の場合のプロセッサにおけるタスクの処理状態を時系列的に示す図である。本図では、すぐにコマ落ちが生じてしまい、ゲームとしての機能を達成できなくなる。このように、動作周波数が低下した状態で、通常時の動作周波数と同じようにタスクを実行すると、リアルタイムタスクのリアルタイム性を保証することは困難になる。

【００１９】

従来、コンピュータの一般的な使われ方では、ワープロソフトやメールソフトが並行して動いている程度で、あまりタスクのリアルタイム性を意識する必要がなかった。しかし、汎用的な機能を併せ持つゲーム機が普及し始め、しかもプロセッサの熱の問題からプロセッサの動作周波数を落とすスロットリング技術が必要になったことから、本発明者は前述のＣＧ映像のコマ落ちといった、過去には存在しなかった問題を認識するに至った。

【００２０】

（実施の形態１）

実施の形態１および実施の形態２では、ノンプリエンプティブなタスク管理がなされる場合、すなわちタスクの切り替えがタスクの自主性に委ねられている場合におけるタスク管理方法について説明する。実施の形態１に係るタスク管理方法は、処理の単位時間をリアルタイム性を保証する保護帯域とリアルタイム性を保証しない非保護帯域に分割し、プロセッサの処理能力が低下したとき、非保護領域において実行すべきタスクの実行を適宜スキップするものである。すなわち、プロセッサの発熱を抑える目的で動作周波数を低くした場合、非保護帯域において実行すべきタスクをベストエフォートで処理する代わりに、保護帯域で実行すべきタスクのリアルタイム性を保証するものである。

【００２１】

「保護帯域」は、単位時間において保証すべきプロセッサの実行サイクル数であり、固定値である。「非保護帯域」は、単位時間において保証しないプロセッサの実行サイクル数であり、０から所定の値まで所定の範囲を有する変動値である。プロセッサの処理能力が最大のとき、その実行サイクル数は、保護帯域における実行サイクル数と非保護帯域における最大の実行サイクル数との合計値になる。例えば、プロセッサの熱制御のために動作周波数を低くする場合、実行サイクル数の変動が非保護帯域においてとりうる実行サイ

ル数の範囲に収まるように動作周波数を低くする。逆にいえば、この範囲で動作周波数を制御すれば、保護帯域で実行すべきタスクのリアルタイム性が保証される。

【００２２】

実施の形態１に係るタスク管理方法は、リアルタイムタスクと非リアルタイムタスクとが混在する際に、リアルタイムタスクを保護帯域に割り当て、非リアルタイムタスクを非保護帯域に割り当ててタスクを実行するものである。そして、プロセッサの熱制御のために行う動作周波数の制御は、動作周波数の制御にともなう実行サイクル数の変動が非保護帯域に収まる範囲で行われる。これにより、プロセッサの熱制御のために行う動作周波数の制御を、プログラム設計とは切り離して行うことができる。つまり、プログラム設計の自由度を制限することなく、柔軟に熱制御を行うことができる。したがって、プロセッサの処理能力を十分に発揮させることができ、かつ動作周波数を変えることによる熱制御も効率的に行うことができる。

【００２３】

図２（ａ）は、動作周波数がf_0の場合、すなわちプロセッサの発熱を抑える必要がない通常の動作周波数のときのタスクの実行状態を示す図である。動作周波数がf_0の場合、それぞれのフレーム期間においてＡ、Ｂ、Ｃの３つのタスクが実行される。タスクＡは、保護帯域に割り当てられるべきリアルタイムタスクである。タスクＢおよびＣは、非保護帯域に割り当てられるべき非リアルタイムタスクである。

【００２４】

図２（ｂ）は、動作周波数が$0.7f_0$のときのタスクの実行状態を示す図である。リアルタイムタスクＡ２およびＡ３は、時刻Ｔ２ならびにＴ３を過ぎて処理が完了している。これでは、コマ落ちが生じてしまう。

【００２５】

図２（ｃ）は、本実施の形態に係るタスク管理方法に基づいてタスクを管理した場合の、動作周波数が$0.7f_0$のときのタスクの実行状態を示す図である。本図では、非リアルタイムタスクＣ１がスキップされている。このように非リアルタイムタスクを実行しないことで、後続のリアルタイムタスクを実行するタイミングが早まる。これにより、リアルタイムタスクのリアルタイム性を保証できる。すなわち本実施の形態では、保護帯域が$0.7f_0$に設定されているので、非保護帯域が０になる。このため、$0.7f_0$を境界にタスク管理の制御が切り替わっている。

【００２６】

図３は、図２（ｃ）を用いて説明したタスク管理方法を利用してタスクを実行するプロセッサシステム１０の構成図である。このプロセッサシステム１０がゲーム機に搭載される。プロセッサシステム１０は、プロセッサである半導体集積回路１００とメインメモリ１４とを含み、これらはバス１２に接続されている。バス１２は、アドレスバス、データバスおよびコントロールバスを含む。半導体集積回路１００は、主処理部１２０、内部クロック生成部１３０、タスク管理部１５０、および周波数制御部１１０を有する。メインメモリ１４は、半導体集積回路１００で実行されるプログラム１６と、各プログラム１６を実行することにより得られた演算結果１８とを保持する。プログラム１６が半導体集積回路１００に読み込まれることによりタスクが実行される。

【００２７】

半導体集積回路１００は、例えば、図示しないキャッシュメモリ、命令レジスタ、演算レジスタ、デコーダ、制御部、演算器などの回路を含み、それらの回路を利用してタスクを実行する。キャッシュメモリまたはメインメモリ１４に記憶されている命令は、フェッチされ命令レジスタに取り込まれる。デコーダは命令レジスタに保持された命令をデコードしてオペコードに対応する制御信号を制御部に供給する。制御部は、例えば制御信号に基づいて、そのオペコードに対する処理を実行するための演算器を選択し、オペランドで指定されたアドレスから演算に必要なデータを取り込み演算レジスタに書き込む。演算器は、例えば演算レジスタに保持されているデータを利用して演算処理を行い、オペランドに指定されたアドレスに書き込む。

【００２８】
本図およびこれから説明する各図は、ハードウエア単位の構成ではなく、機能単位の構成を示している。また、各機能ブロックは、同時に半導体集積回路１００の内部に静的に形成されている必要はなく、一定の期間に動的に形成されればよい。本実施の形態では、タスク管理部１５０および周波数制御部１１０は、例えばオペレーティングシステムに組み込まれたプログラムが実行されることにより、半導体集積回路１００に動的に形成されるものとする。また、主処理部１２０は本来は半導体集積回路１００全体と考えることもできるが、ここでは理解を助けるために、周波数制御部１１０、内部クロック生成部１３０、タスク管理部１５０を除く機能部分を指すものとする。
【００２９】
ベースクロック供給部３０は、半導体集積回路１００に対してベースクロックを供給する。内部クロック生成部１３０は、例えばＰＬＬ（Phase Locked Loop）を含み、ベースクロックの周波数の整数倍のクロックを生成する。内部クロック生成部１３０が生成するクロックを動作クロックとよび、動作クロックの周波数を動作周波数とよぶ。半導体集積回路１００に含まれる各回路は、動作クロックの上昇または下降のタイミングで動作する。内部クロック生成部１３０は、ＰＬＬを構成する回路に含まれるカウンタのカウンタ値を調整することで動作周波数を変えることができる。
【００３０】
温度センサ１０２は、主処理部１２０またはその周辺回路の温度を測定し、測定した温度を周波数制御部１１０に出力する。温度センサ１０２は、半導体集積回路１００の外部に設けられてもよいし、半導体集積回路１００の内部、すなわちダイ上に設けられてもよい。
【００３１】
周波数制御部１１０は、半導体集積回路１００の温度に応じて、熱制御のために必要な動作周波数を割り出し、その周波数で動作クロックを生成するように内部クロック生成部１３０を制御する。周波数制御部１１０は、温度が所定の閾値より高くなった場合に、動作周波数を低くするように内部クロック生成部１３０を制御する。動作周波数を低くすることにより、半導体集積回路１００における発熱量を抑えることができ、さらにヒートシンクなどの放熱機構が作用して半導体集積回路１００の温度を下げることができる。
【００３２】
熱制御のために調整する動作周波数の範囲は、保護帯域より低くならないことが好ましく、半導体集積回路１００の温度に応じて段階的に動作周波数を調整してもよい。実行すべきタスクが少ないときや半導体集積回路１００の発熱量が大きく非保護帯域だけでは温度を十分に下げることが出来ない非常事態と判断した場合、周波数制御部１１０は、動作周波数を保護帯域より低くしてもよい。こうした、発熱量に対する動作周波数の制御方法はいろいろあり、周波数制御部１１０は任意の方法に基づいて動作周波数を割り出してよい。
【００３３】
例えば、周波数制御部１１０は、半導体集積回路１００の温度と動作周波数とを対応付けたテーブルを参照して動作周波数を割り出す。そして、周波数制御部１１０は、内部クロック生成部１３０にその動作周波数の動作クロックを生成させる。
【００３４】
主処理部１２０は、タスク管理部１５０に指示されたタスクに対応するプログラム１６をメインメモリ１４から読み込み、タスクを実行する。そして、主処理部１２０は、そのタスクを実行することにより得た演算結果１８をメインメモリ１４に書き込む。
【００３５】
タスク管理部１５０は、周波数制御部１１０から主処理部１２０の動作周波数を特定する周波数情報を受け付け、その情報に基いて前述したタスク管理方法に沿ってタスクのスケジューリングを行う。
【００３６】

図４は、図３のスケジュール管理部１５０の内部構成図である。周波数検出部１５２は、動作周波数を検出する。本実施の形態では、周波数検出部１５２は、周波数制御部１１０から周波数情報を受け付けることで動作周波数を把握する。周波数検出部１５２は、周波数情報を切替指示部１５４に出力する。

【００３７】

切替指示部１５４は、スケジュール作成部１５６および指示部１５８を有し、主処理部１２０に実行させるタスクの切り替え指示を行う。詳細は後述するが、スケジュール作成部１５６は、制御目標テーブル１６０とタスクテーブル１６４を参照して動作周波数に応じたタスクのスケジューリングを行う。指示部１５８は、スケジュール作成部１５６で生成されたスケジュールに基いて、それぞれのタスクの実行指示を主処理部１２０に対して行う。

【００３８】

図５は、図３の制御目標テーブル１６０に保持されているデータをグラフ表示した図である。制御目標テーブル１６０は、動作周波数と、非リアルタイムタスクをどの程度実行するかを決める制御目標とを対応付けて保持する。この制御目標は、リアルタイムタスクの実行回数に対する非リアルタイムタスクの実行回数の割合である。この割合を、以下「実行率」とよぶ。制御目標テーブル１６０に保持されているこれらのデータを、横軸に動作周波数をとり、縦軸に非リアルタイムタスクの実行率をとってグラフにすると、本図のように０．７f_0を境界として異なる様相になる。本実施の形態では、保護帯域が０．７f_0に設定されている。非保護帯域であるf_0から０．９f_0のときは、非リアルタイムタスクの実行率が１００％であり、０．９f_0から０．７f_0のときは、１００％から１０％まで非リアルタイムタスクの実行率が線形に減少される。また、動作周波数が、保護帯域である０．７f_0より低くなると非リアルタイムタスクの実行率が１０％になる。０．７f_0より低くなった場合、リアルタイムタスクのリアルタイム性を保証するという意味で、非リアルタイムタスクの実行率を０％にしてもよいが、本実施の形態では非リアルタイムタスクが全く実行されないという状況を避けるために実行率が１０％になっている。例えば、実行率「３０％」は、リアルタイムタスクを１０回実行するのに対して、非リアルタイムタスクを３回実行することを示す。図３のスケジュール作成部１５６は、この制御目標に基づいて非リアルタイムタスクを実行するタイミングを調整する。

【００３９】

図６は、図４のタスクテーブル１６４のデータ構造の一例を示す図である。タスクテーブル１６４は、属性欄１８４とタスク欄１８６と優先度欄１８８とを対応付けて有する。属性欄１８４は、タスクの属性を示す属性情報を保持する。本実施の形態では、属性としてリアルタイムタスクと非リアルタイムタスクとがある。本図で「ＲＴ」は、リアルタイムタスクを示し、「ＮＲＴ」は非リアルタイムタスクを示す。タスク欄１８６は、例えばプログラムのファイル名やタスクの識別情報などタスクを特定するための情報を保持する。本図で、例えば「レンダリング」タスクはリアルタイムタスクとして登録され、「ブラウザ」タスクは非リアルタイムタスクとして登録されている。優先度欄１８８は、タスクの優先度を示す情報を保持する。本図では、優先度は１０段階で示されており、「１０」が最も優先度が高く、「１」が最も優先度が低い。

【００４０】

図４に戻り、登録部１６２は、例えばユーザがプログラムの実行指示をした場合に、プログラムコードに含まれる属性情報を解釈し、そのプログラムを特定する情報に対応付けてタスクテーブル１６４に登録する。属性情報がプログラムコードに含まれていない場合、登録部１６２はそのプログラムをリアルタイムタスクとしてタスクテーブル１６４に登録する。これにより、例えば属性情報を含まない古いプログラムも実行できる。

【００４１】

また、別の例では、登録部１６２は、各タスクにて実行されるプログラムの性質を判断し、その判断に基いてそのプログラムの属性を推定してタスクテーブル１６４に登録してもよい。登録部１６２は、そのプログラムに含まれる命令、そのプログラムによるプロセ

ツの占有率や占有時間などの、プログラムを実行することにより間接的に得られる特徴に基いて属性を推定する。例えば、登録部１６２は、ノンプリエンプティブなタスク管理がなされる場合は、一定期間におけるプロセッサの占有時間をタスク毎に測定し、占有時間の長いタスクをリアルタイムタスクとして推定し、占有時間の短いタスクを非リアルタイムタスクとして推定してもよい。これにより、プログラムコードに属性情報が含まれていないプログラムに対しても、適切なタスク管理を行うことができる。

【００４２】

図７は、図４のスケジュール作成部１５６の内部構成図である。第１計画部１７０は、タスクテーブル１６４を参照して、リアルタイムタスクの実行順序を決める。リアルタイムタスクが複数ある場合、第１計画部１７０は、図６の優先度欄１８８を参照して、優先度の高いリアルタイムタスクが先になるように実行順序を決める。そして、第１計画部１７０は、リアルタイムタスクの実行順序を統合部１７８に出力する。

【００４３】

第２計画部１７２は、タスクテーブル１６４を参照して、非リアルタイムタスクの実行順序を決め、実行順序を統合部１７８に出力する。第２計画部１７２は、設定部１７４、第１カウンタ１７６ａ、および第２カウンタ１７６ｂを有する。第１カウンタ１７６ａおよび第２カウンタ１７６ｂを、総称してカウンタ１７６とよぶ。カウンタ１７６は、非リアルタイムタスクのスケジューリング処理を行うたびに計数を行い、設定された割合で非リアルタイムタスクの実行を許可する。

【００４４】

例えば第１カウンタ１７６ａは、２５％の割合で非リアルタイムタスクの実行を許可するカウンタであり、リアルタイムタスクを３回実行するのに対して、非リアルタイムタスクを１回実行することを許可する。また、許可のタイミングは、任意に決めることができる。本図では、第１カウンタ１７６ａの許可タイミングは「○××××」として図示されている。「○」が許可することを示し、「×」が許可しないことを示すので、第１カウンタ１７６ａは、最初に非リアルタイムタスクの実行を許可した後、３回スキップすることを示す。

【００４５】

設定部１７４は、周波数検出部１５２から現在の動作周波数を受け付け、その動作周波数に応じた非リアルタイムタスクの制御目標を制御目標テーブル１６０から読み込む。そして、設定部１７４は、制御目標に応じてカウンタ１７６が非リアルタイムタスクの実行を許可する割合を設定する。例えば、２５％で非リアルタイムタスクを実行する場合、設定部１７４は、カウンタ１７６に４回に１回の割合で非リアルタイムタスクの実行を許可するように設定する。

【００４６】

非リアルタイムタスクが複数ある場合、設定部１７４は、非リアルタイムタスク毎にカウンタ１７６を設け、複数の非リアルタイムタスクの実行回数を合計した値が、制御目標になるようにそれぞれのカウンタ１７６の設定を行う。例えば、非リアルタイムタスクが２つあり、２５％の割合で非リアルタイムタスクを実行する場合、設定部１７４は、第１カウンタ１７６ａおよび第２カウンタ１７６ｂのそれぞれにリアルタイムタスクを８回実行するのに対して、非リアルタイムタスクの実行を１回許可するように設定する。更に、設定部１７４は、２つの非リアルタイムタスクを実行するタイミングが重ならないように、許可タイミングを設定する。つまり、第１カウンタ１７６ａの許可タイミングを「×○××××××」とし、第２カウンタ１７６ｂの許可タイミングを「××××××○×」とする。

【００４７】

また、例えば５０％の割合で非リアルタイムタスクを実行する場合、設定部１７４は、許可タイミングを「○○××」ではなく、「○×○×」のように「○」と「×」を離して、すなわち非リアルタイムタスクの実行を許可するタイミングを分散するように許可タイミングを設定する。このように非リアルタイムタスクの実行を許可するタイミングを分散

することで、非リアルタイムタスクがよりスムーズに実行される。

【００４８】

統合部１７８は、第１計画部１７０から供給されるリアルタイムタスクの実行順序と、第２計画部１７２から供給される非リアルタイムタスクの実行順序とを、リアルタイムタスクの実行順序が先になるように並べてスケジュールを作成する。そして、統合部１７８は、作成したスケジュールを指示部１５８に出力する。

【００４９】

図８は、図４のタスク管理部１５０におけるタスクをスケジューリングする処理のフローチャートである。図４の周波数検出部１５２は、動作周波数を検出する（Ｓ１０）。図４のスケジュール作成部１５６は、動作周波数に基づいて図４の制御目標テーブル１６０を参照して、リアルタイムタスクと非リアルタイムタスクの実行順序をスケジューリングする（Ｓ１２）。図４の指示部１５８は、スケジュール作成部１５６にスケジューリングされた順序でタスクの実行指示を行う（Ｓ１４）。

【００５０】

（実施の形態２）

図９は、実施の形態２に係るタスク管理部１５０の内部構成図である。実施の形態２は、半導体集積回路１００の使用率に応じて制御目標テーブル１６０を最適化する形態である。既に説明した構成と同一の符号を付した本図の構成は、機能および動作が既に説明した構成と略同一である。以下、既に説明した構成における機能とは異なる点を中心に説明する。

【００５１】

プロセッサ使用率検出部１９０は、図３の半導体集積回路１００の使用率を例えばフレーム期間ごとに検出する。そして、プロセッサ使用率検出部１９０は、使用率を更新部１９２に出力する。更新部１９２は、所定の期間における使用率の平均値を算出し、その平均値に基いて、非リアルタイムタスクの実行率を増やすように制御目標テーブル１６０を最適化する。例えば更新部１９２は、０．５秒間すなわち３０フレーム分の使用率の平均値を算出する。そして、平均値が閾値より低い場合、すなわち負荷が比較的少ない場合、更新部１９２は制御目標テーブル１６０における非リアルタイムタスクの実行率を上げる。平均値が閾値より高い場合、更新部１９２は制御目標テーブルをデフォルト値に戻す。閾値や非リアルタイムタスクの実行率の上げ幅は、実験により適切な値が設定されてよいし、実行にともない徐々に反映されてもよい。

【００５２】

図１０は、図９の更新部１９２により最適化された制御目標テーブル１６０に保持されているデータをグラフ表示した図である。本図では、保護帯域が０．５f_0までシフトされ、０．９f_0から０．５f_0のときは、１００％から１０％まで非リアルタイムタスクの実行率が線形に減少される。このように、プロセッサの使用率に応じて保護帯域を調整することにより、プロセッサの処理能力を有効に利用することができる。

【００５３】

図１１は、図９のタスク管理部１５０におけるタスクをスケジューリングする処理のフローチャートである。図９のタスク管理部１５０におけるスケジューリング処理は、図８を用いて説明した図４のタスク管理部１５０におけるスケジューリング処理に、制御目標テーブルを最適化する処理（Ｓ２０）を加えることで実現される。

【００５４】

図１２は、図１１の制御目標テーブルを最適化する処理（Ｓ２０）の詳細なフローチャートである。図９のプロセッサ使用率検出部１９０は、図３の半導体集積回路１００の使用率を検出する（Ｓ２２）。図９の更新部１９２は、所定の期間が経過したか否かを判定する（Ｓ２４）。所定の期間が経過した場合（Ｓ２４のＹ）、更新部１９２はその期間における使用率の平均値を算出する（Ｓ２６）。そして、使用率が所定の閾値より高い場合（Ｓ２８のＹ）、デフォルトの制御目標テーブルに戻す（Ｓ３４）。ステップ２８で、使用率が所定の閾値より低い場合（Ｓ２８のＮ）、更新部１９２は制御目標テーブルを非リ

アルゴリズムへの実行率を上げるように変更する（Ｓ３０）。ステップ２４で、所定の期間が経過していない場合（Ｓ２４のＮ）、図１１のステップ１０に進む。このように制御目標テーブル１６０を主処理部１２０の使用率に応じて最適化することにより、主処理部１２０の処理能力を引き出すことができる。

【００５５】

（実施の形態３）

実施の形態３および実施の形態４では、プリエンプティブなタスク管理がなされている場合、すなわちタイマ割込により強制的にタスクの切り替えが行われる場合におけるタスク管理方法について説明する。以下、フレーム期間を「ｔ」とする。実施の形態３に係るタスク管理方法は、非リアルタイムタスクを、非保護帯域として割り当てられた期間に実行するものである。以下、保護帯域として割り当てられた期間を「保護期間ｔｒ」といい、非保護帯域として割り当てられた期間を「非保護期間ｔｎ」という。複数の非リアルタイムタスクがある場合には、非保護期間ｔｎを等分して、それぞれの非リアルタイムタスクが実行される。

【００５６】

図１３（ａ）は、動作周波数がf_0の場合のタスクの実行状態を示す図である。動作周波数がf_0のとき、保護期間ｔｒは０．７ｔであり、非保護期間ｔｎは０．３ｔである。非リアルタイムタスクが２つあるので、それぞれの非リアルタイムタスクは０．１５ｔの期間ずつ順番に実行される。

【００５７】

図１３（ｂ）は、動作周波数が$0.8f_0$の場合のタスクの実行状態を示す図である。動作周波数が$0.8f_0$のとき、動作周波数の周期はｔ／（０．８・f_0）であるので、保護期間ｔｒは$0.7f_0$・ｔ／（０．８・f_0）＝７／８・ｔであり、非保護期間ｔｎは１／８・ｔである。非リアルタイムタスクが２つあるので、それぞれの非リアルタイムタスクは１／１６・ｔの期間ずつ順番に実行される。本図では、保護期間ｔｒの間リアルタイムタスクＲＴが占有しているが、リアルタイムタスクＲＴのプログラムは保護期間ｔｒより短い期間で処理が完了するように設計されるので、一般的には保護期間ｔｒの全てを占有することはない。

【００５８】

図１４は、実施の形態３に係るタスク管理部１５０の内部構成図の一例を示す図である。既に説明した構成と同一の符号を付した本図の構成は、機能および動作が既に説明した構成と略同一である。以下、既に説明した構成における機能とは異なる点を中心に説明する。スケジュール作成部１５６は、図１３を用いて説明したようにタスクのスケジューリングを行う。スケジュール作成部１５６は、タスクテーブル１６４からリアルタイムタスクと非リアルタイムタスクを読み込み、リアルタイムタスクに続けて非リアルタイムタスクを実行するようにスケジュールを作成する。また、スケジュール作成部１５６は、スケジューリングのために制御目標テーブル１６０を参照して、動作周波数に応じた非リアルタイムタスクの実行時間を割り出す。そして、スケジュール作成部１５６は、例えば、非リアルタイムタスク毎に実行時間を対応付けてスケジュールを作成する。指示部１５８はスケジュールに含まれる実行時間に基いて、非リアルタイムタスクを切り替えるための割込タイマを設定する。

【００５９】

図１５は、図１４の制御目標テーブル１６０に保持されているデータをグラフ表示した図である。制御目標テーブル１６０は、動作周波数と、非リアルタイムタスクを実行するためにプロセッサを占有できる時間、すなわち非保護期間ｔｎとを対応付けて保持する。制御目標テーブル１６０に保持されているこれらのデータを、横軸に動作周波数をとり、縦軸に非保護期間ｔｎをとってグラフにすると、本図のように$0.7f_0$を境界として異なる様相になる。本実施の形態では、保護帯域が$0.7f_0$に設定されている。非保護帯域であるf_0から$0.9f_0$のときは、非保護期間ｔｎが０．３ｔであり、$0.9f_0$から$0.7f_0$のときは、０．３ｔから０．０１ｔまで線形に非保護期間ｔｎが短くなって

いる。また、動作周波数が、保護帯域である$0.7f_0$より低くなると保護期間tnが0．01tになる。0．7f_0より低くなった場合、リアルタイムタスクのリアルタイム性を保証するという意味で、非保護期間tnを0にしてもよいが、本実施の形態では非リアルタイムタスクが全く実行されないという状況を避けるために非保護期間tnが0．01tになっている。例えば、非保護期間tnが「0．1t」であり、非リアルタイムタスクが2つある場合、それぞれの非リアルタイムタスクの実行時間は0．05tになる。図14のスケジュール作成部156は、この制御目標に基づいて非リアルタイムタスクを実行時間を調整する。

【0060】

（実施の形態4）

図16は、実施の形態4に係るタスク管理部150の内部構成図である。実施の形態4は、半導体集積回路100の使用率に応じて制御目標テーブル160を最適化する形態である。既に説明した構成と同一の符号を付した本図の構成は、機能および動作が既に説明した構成と略同一である。以下、既に説明した構成における機能とは異なる点を中心に説明する。

【0061】

更新部192は、所定の期間における半導体集積回路100の使用率の平均値を算出し、その平均値に基いて、非保護期間tnを増やすように制御目標テーブル160を最適化する。平均値が閾値より小さい場合、すなわち負荷が比較的少ない場合、更新部192は制御目標テーブル160における非保護期間tnを長くする。平均値が閾値より高い場合、更新部192は制御目標テーブルをデフォルト値に戻す。更新の判断に利用する閾値や非保護期間tnを延長する長さは、実験により適切な値が設定されてよいし、実行にともない徐々に反映されてもよい。

【0062】

図17は、図16の更新部192により最適化された制御目標テーブル160に保持されているデータをグラフ表示した図である。本図では、保護帯域が$0.5f_0$までシフトされ、$0.9f_0$から$0.5f_0$のときは、0．3tから0．01tまで線形に非保護期間tnが短くなっている。このように、プロセッサの使用率に応じて保護帯域を調整することにより、プロセッサの処理能力を有効に利用することができる。

【0063】

以上、本発明を実施の形態をもとに説明した。これらの実施の形態は例示であり、それらの各構成要素や各処理プロセスの組合せにいろいろな変形例が可能なこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。こうした変形例として、実施の形態1から実施の形態4では、1フレーム期間に描画処理を完了すべき時間的制約のあるタスクを保護帯域に配し、時間的制約のないタスクを非保護帯域に配することとして説明したが、保護帯域または非保護帯域に配する条件は、これに限定されない。

【0064】

時間的制約とは異なる観点からタスクを保護帯域または非保護帯域に配してよく、例えば、データを確実に記録する必要があるか否かという観点から、タスクを保護帯域または非保護帯域のいずれかに配してもよい。この場合、例えば番組放送を録画するタスクとワードプロセッサのタスクがある場合、録画のタスクを、データを確実に記録するという観点から保護帯域に配し、ワードプロセッサのタスクを非保護帯域に配してもよい。このように、保護帯域と非保護帯域に分配する条件を変えることにより、実施の形態で説明したタスク管理方法を、例えば航空機の制御用コンピュータや自動車の制御用コンピュータなどリアルタイムOSを搭載する電子装置に採用することができる。

【0065】

別の変形例として、実施の形態では熱制御という観点から半導体集積回路100の動作周波数が制御されることとしたが、例えば、電力消費という観点から動作周波数が制御されてもよい。消費電力は、クロックの動作周波数、回路のトランジスタ数、および電源電

圧の2乗に比例する。逆にいえば、動作周波数、負荷となるトランジスタ数、および電源電圧が分かれば、消費電力を算出することができる。例えば、図３の周波数制御部１１０は、レギュレータの出力電圧値を測定するセンサから電圧値を取得して、その電圧値から算出される消費電力が所定の閾値より高くなった場合、熱暴走を防止するために、動作周波数を低くしてもよい。また、電源モードがバッテリモードかＡＣ電源モードかを判断してバッテリモードの場合、節電のために、動作周波数を低くしてもよい。こうした場合でも、実施の形態で説明したタスク管理方法によれば、リアルタイムタスクのリアルタイム性が保証される。

【００６６】

別の変形例として、実施の形態では主処理部１２０における動作周波数に応じて非リアルタイムタスクの実行率が調整されることとしたが、主処理部１２０またはその周辺回路の温度に応じて非リアルタイムタスクの実行率が調整されてもよい。例えば、主処理部１２０に供給するクロックの動作周波数を、主処理部１２０またはその周辺回路の温度に応じてソフトウエアとは独立して制御する回路が設けられている場合、動作周波数は主処理部１２０またはその周辺回路の温度に応じてハードウエア的に制御される。こうした回路において、ソフトウエア的に取得できる情報が周波数情報ではなく温度情報の場合、すなわち図３のタスク管理部１５０が周波数情報ではなく温度情報を取得できる場合、温度情報に基いてタスクの実行率が調整されてもよい。この場合、例えば図４の制御目標テーブル１６０は、主処理部１２０またはその周辺回路の温度と実行率とを対応付けて保持し、図４のスケジュール作成部１５６はこのテーブルに基いて非リアルタイムタスクを実行するタイミングを調整する。

【００６７】

また、主処理部１２０または半導体集積回路１００の消費電力に応じて非リアルタイムタスクの実行率が調整されてもよい。各種の電源管理ソフトウエア等から図３のタスク管理部１５０が周波数情報ではなく消費電力情報を取得できる場合、消費電力情報に基いてタスクの実行率が調整されてもよい。この場合、例えば図４の制御目標テーブル１６０は、消費電力情報と実行率とを対応付けて保持し、図４のスケジュール作成部１５６はこのテーブルに基いて非リアルタイムタスクを実行するタイミングを調整する。

【図面の簡単な説明】

【００６８】

【図１】（ａ）は、通常時の動作周波数で動作するプロセッサにおけるタスクの処理状態を時系列的に示す図であり、（ｂ）は、動作周波数が低下した場合のプロセッサにおけるタスクの処理状態を時系列的に示す図であり、（ｃ）は、更に動作周波数が低下した場合のプロセッサにおけるタスクの処理状態を時系列的に示す図である。

【図２】（ａ）は、動作周波数がf_0の場合、すなわちプロセッサの発熱を抑える必要がない通常の動作周波数のときのタスクの実行状態を示す図であり、（ｂ）は、動作周波数が０．７f_0のときのタスクの実行状態を示す図であり、（ｃ）は、本実施の形態に係るタスク管理方法に基づいてタスクを管理した場合の、動作周波数が０．７f_0のときのタスクの実行状態を示す図である。

【図３】図２を用いて説明したタスク管理方法を利用してタスクを実行するプロセッサシステムの構成図である。

【図４】図３のタスク管理部の内部構成図である。

【図５】図３の制御目標テーブルに保持されているデータをグラフ表示した図である。

【図６】図４のタスクテーブルのデータ構造の一例を示す図である。

【図７】図４のスケジュール作成部１５６の内部構成図である。

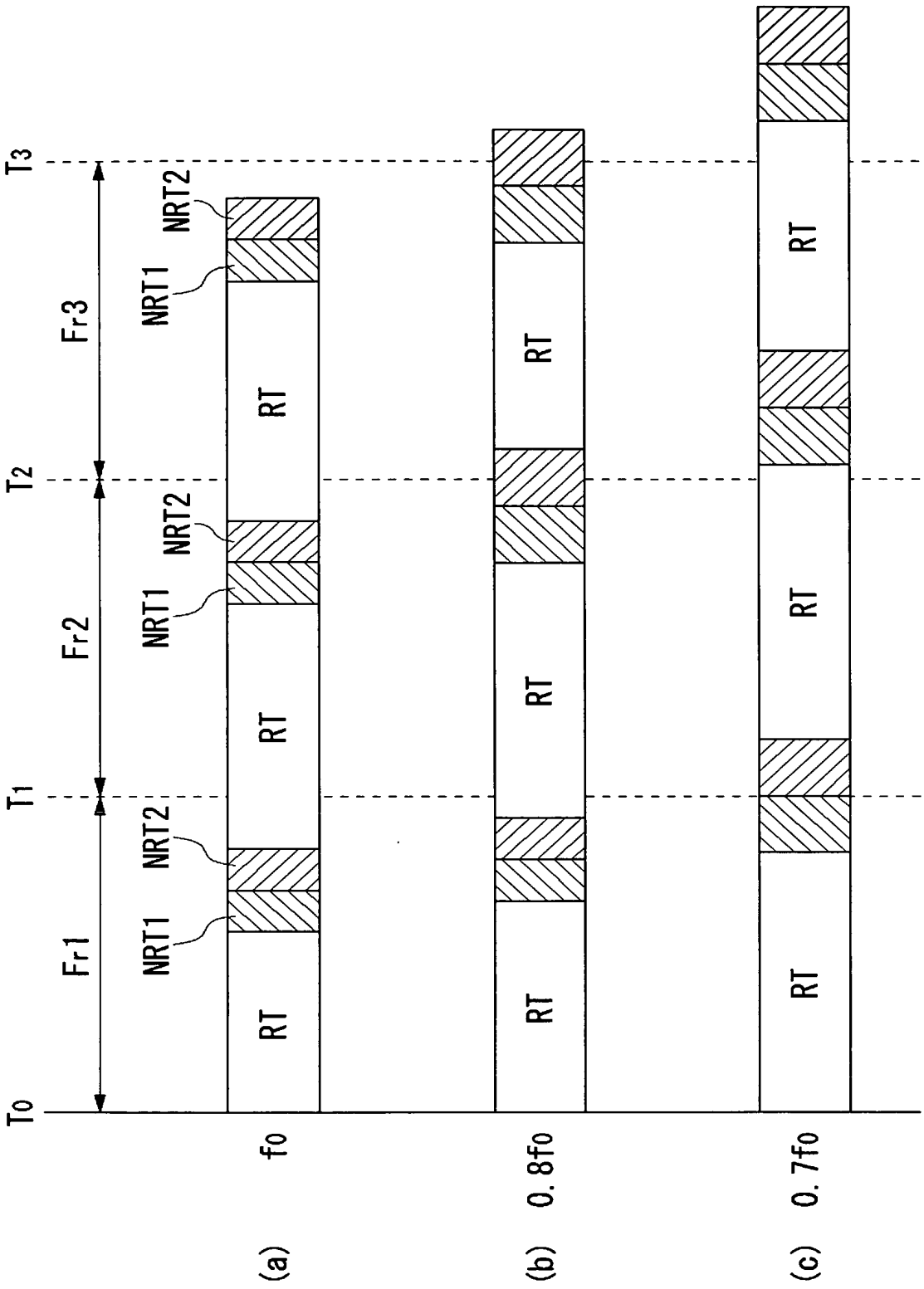
【図８】図４のタスク管理部におけるタスクを管理する処理のフローチャートである。

【図９】実施の形態２に係るタスク管理部の内部構成図である。

【図１０】図９の更新部により最適化された制御目標テーブルに保持されているデー

タをグラフ表示した図である。
【図１１】図９のタスク管理部におけるタスク管理処理のフローチャートである。
【図１２】図９の制御目標テーブル更新処理の詳細なフローチャートである。
【図１３】（ａ）は、動作周波数がf_0の場合のタスクの実行状態を示す図であり、（ｂ）は、動作周波数が０．８f_0の場合のタスクの実行状態を示す図である。
【図１４】実施の形態３に係るタスク管理部の内部構成図の一例を示す図である。
【図１５】図１４の制御目標テーブルに保持されているデータをグラフ表示した図である。
【図１６】実施の形態４に係るタスク管理部の内部構成図である。
【図１７】図１６の更新部により最適化された制御目標テーブルに保持されているデータをグラフ表示した図である。
【符号の説明】
【００６９】
１０　プロセッサシステム、１２　バス、１４　メインメモリ、１６　プログラム、１８　演算結果、３０　ベースクロック供給部、１００　半導体集積回路、１０２　温度センサ、１１０　周波数制御部、１２０　主処理部、１３０　内部クロック生成部、１５０　タスク管理部、１５２　周波数検出部、１５４　切替指示部、１５６　スケジュール作成部、１５８　指示部、１６０　制御目標テーブル、１６２　登録部、１６４　タスクテーブル、１７０　第１計画部、１７２　第２計画部、１７４　設定部、１７６　カウンタ、１９０　プロセッサ使用率検出部、１９２　更新部。

【書類名】図面
【図１】



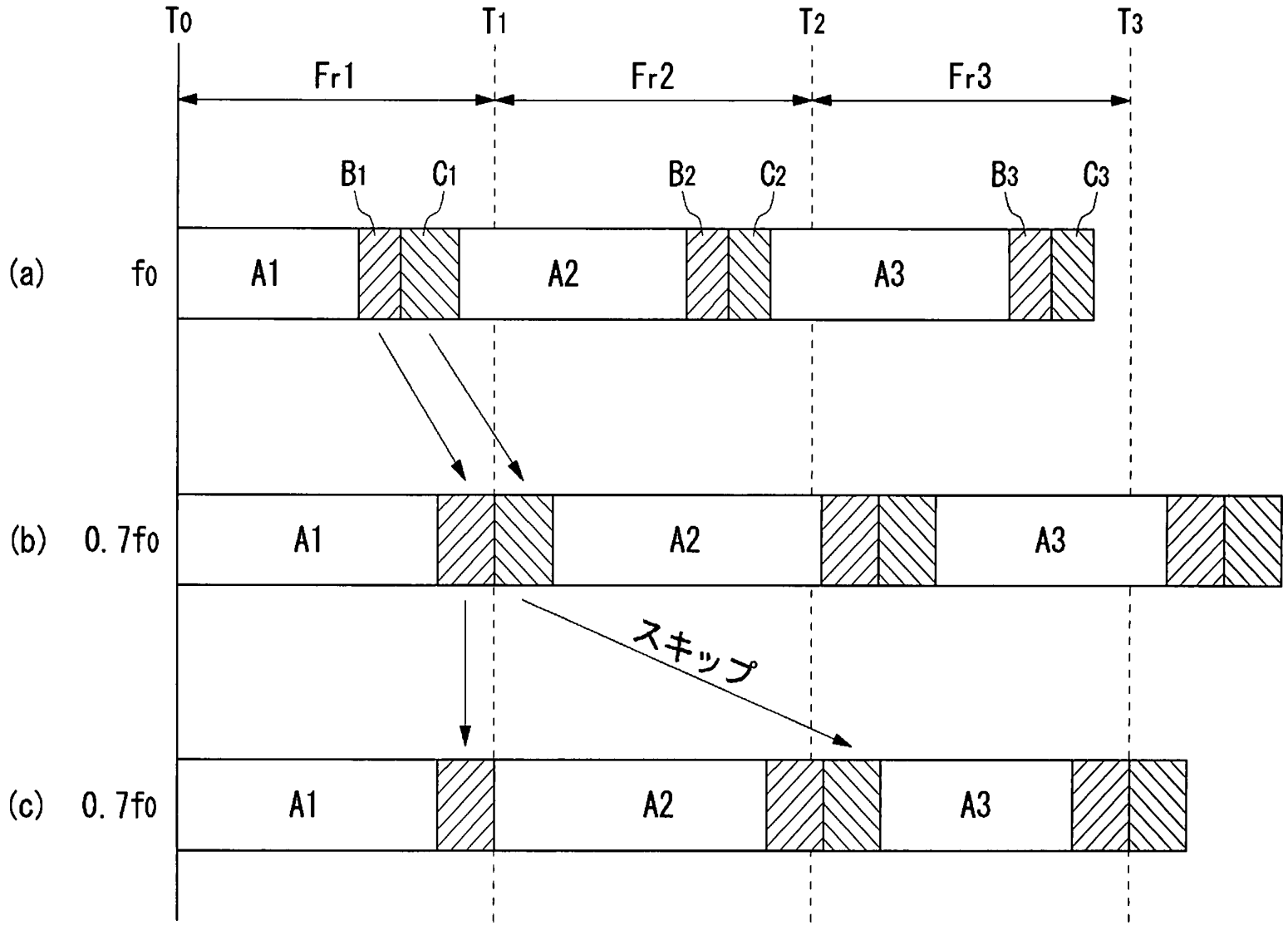
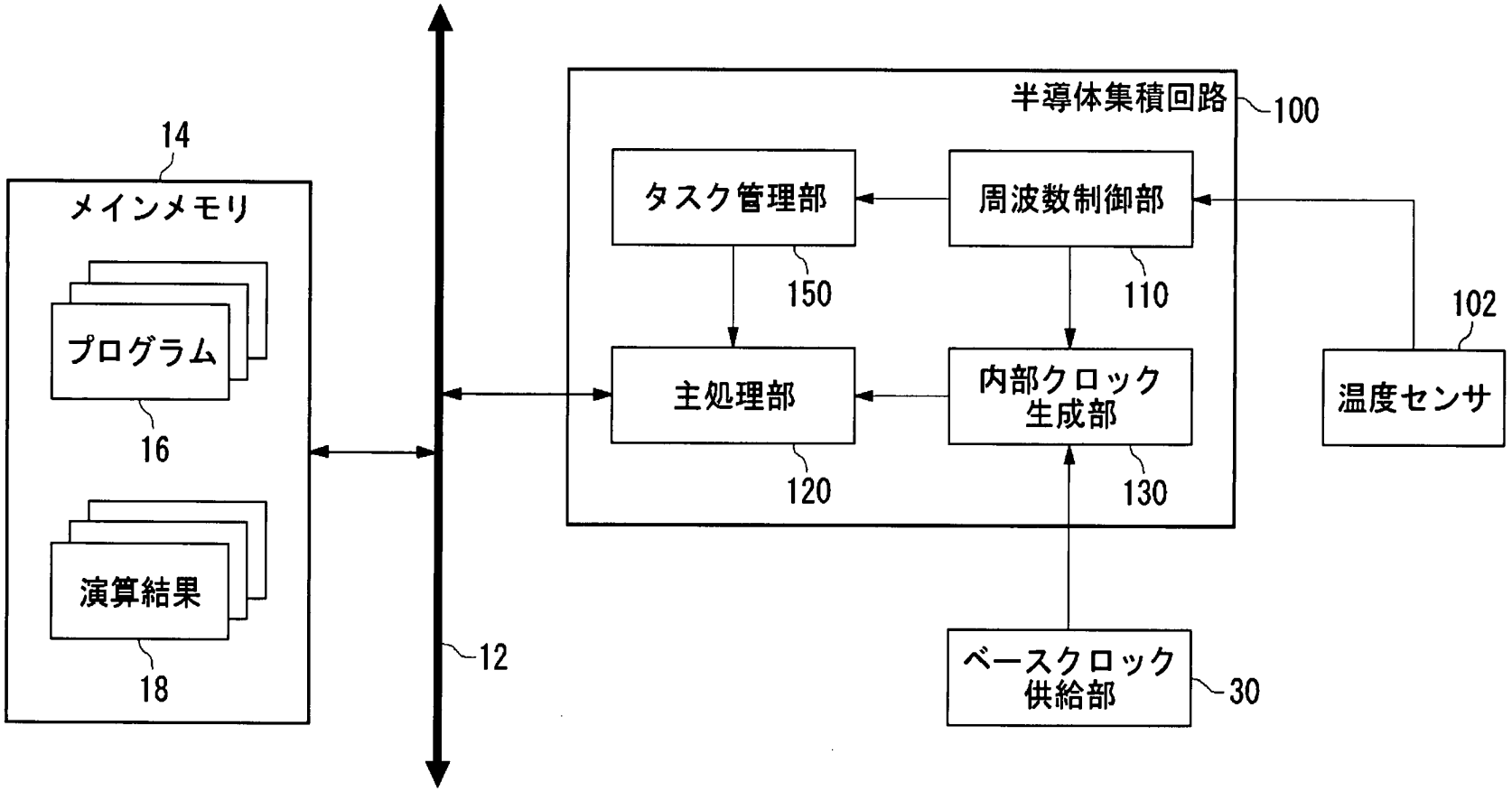
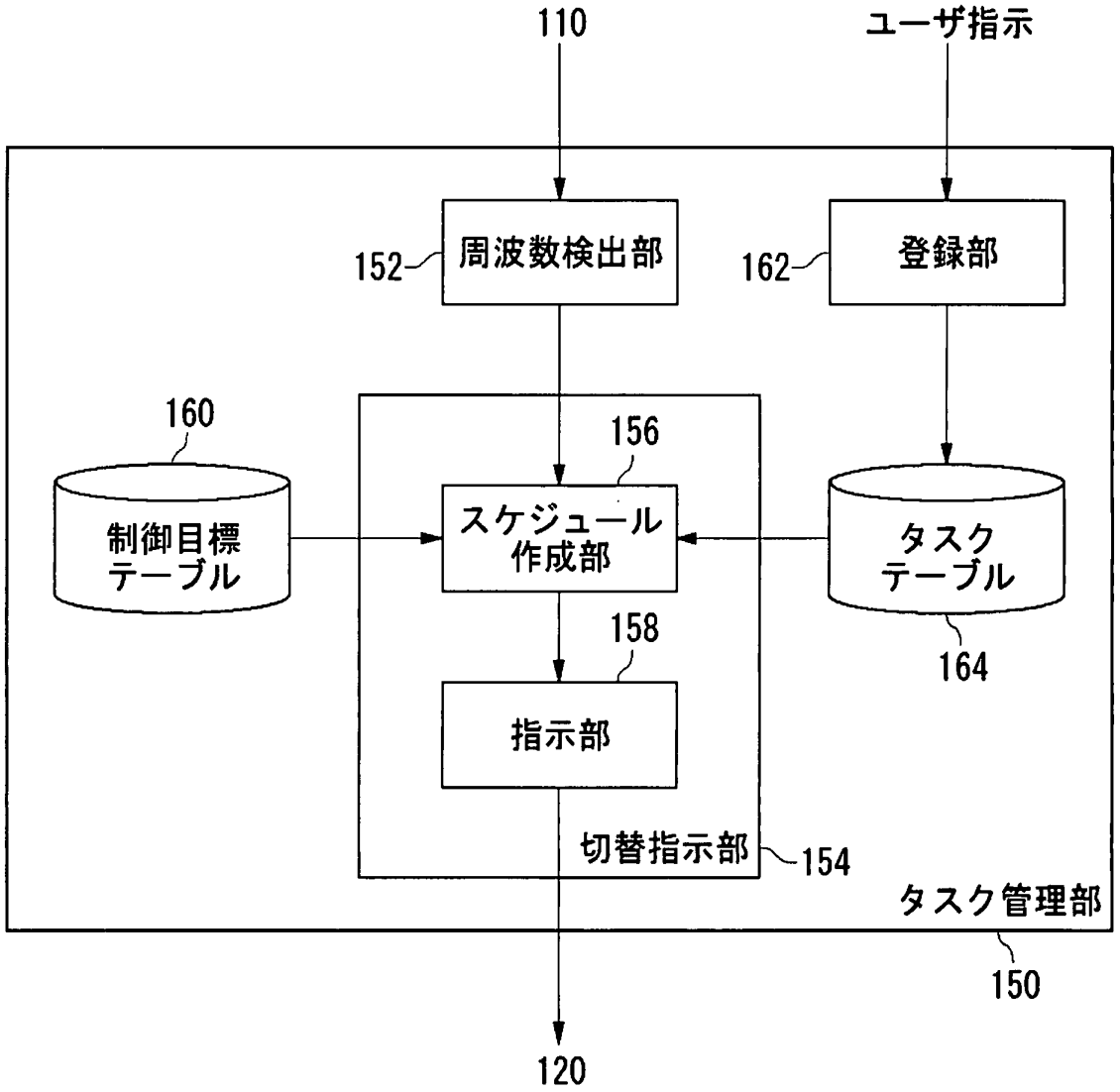

T0
T1
T2
T3
Fr1
Fr2
Fr3
B1
C1
B2
C2
B3
C3
(a)
f0
A1
A2
A3
(b)
0.7f0
A1
A2
A3
スキップ
(c)
0.7f0
A1
A2
A3

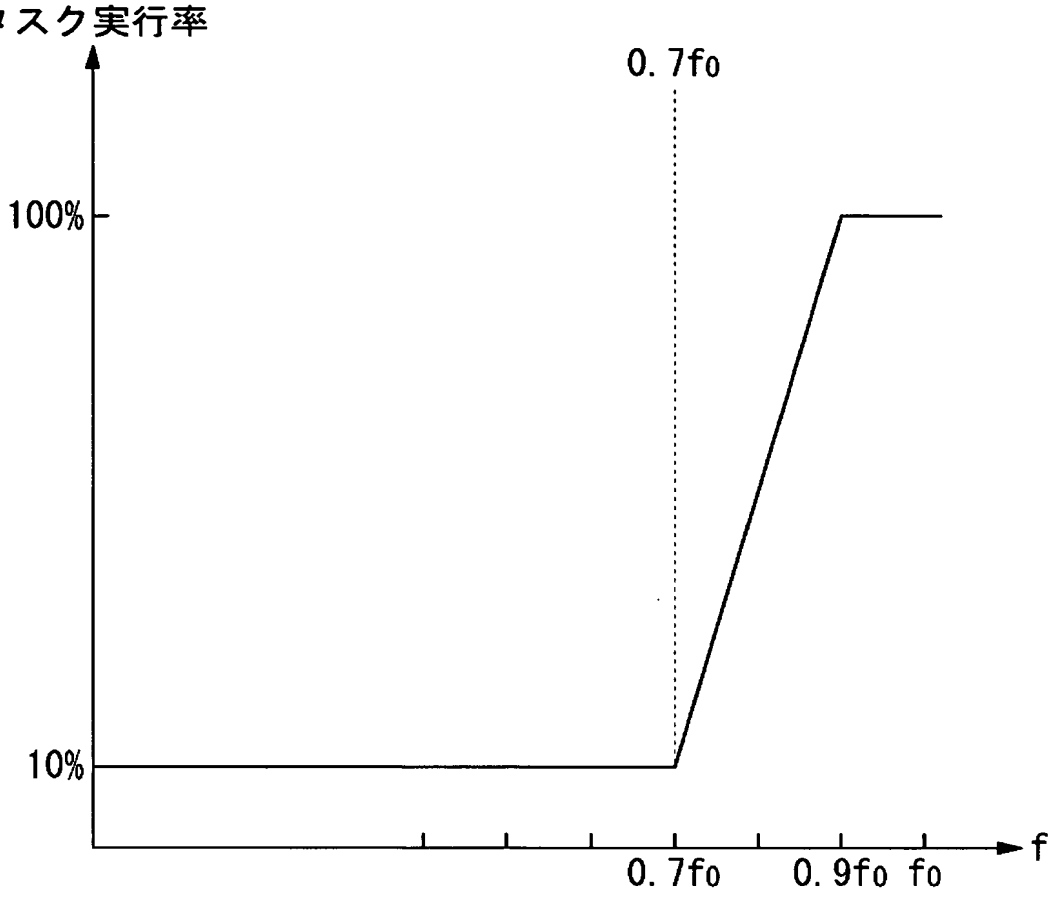
図 7-1


半導体集積回路
100
14
メインメモリ
プログラム
16
演算結果
18
タスク管理部
150
周波数制御部
110
主処理部
120
内部クロック
生成部
130
102
温度センサ
12
ベースクロック
供給部
30

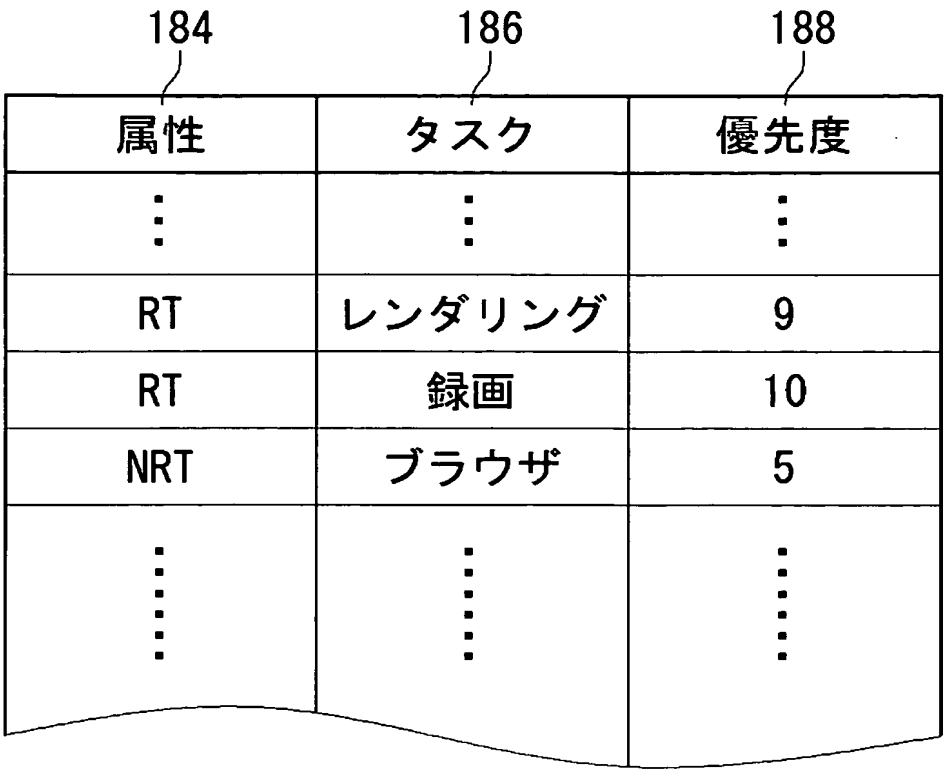
10


110
ユーザ指示
152
周波数検出部
162
登録部
160
156
制御目標
テーブル
スケジュール
作成部
タスク
テーブル
158
164
指示部
切替指示部
154
タスク管理部
150
120

【図３】

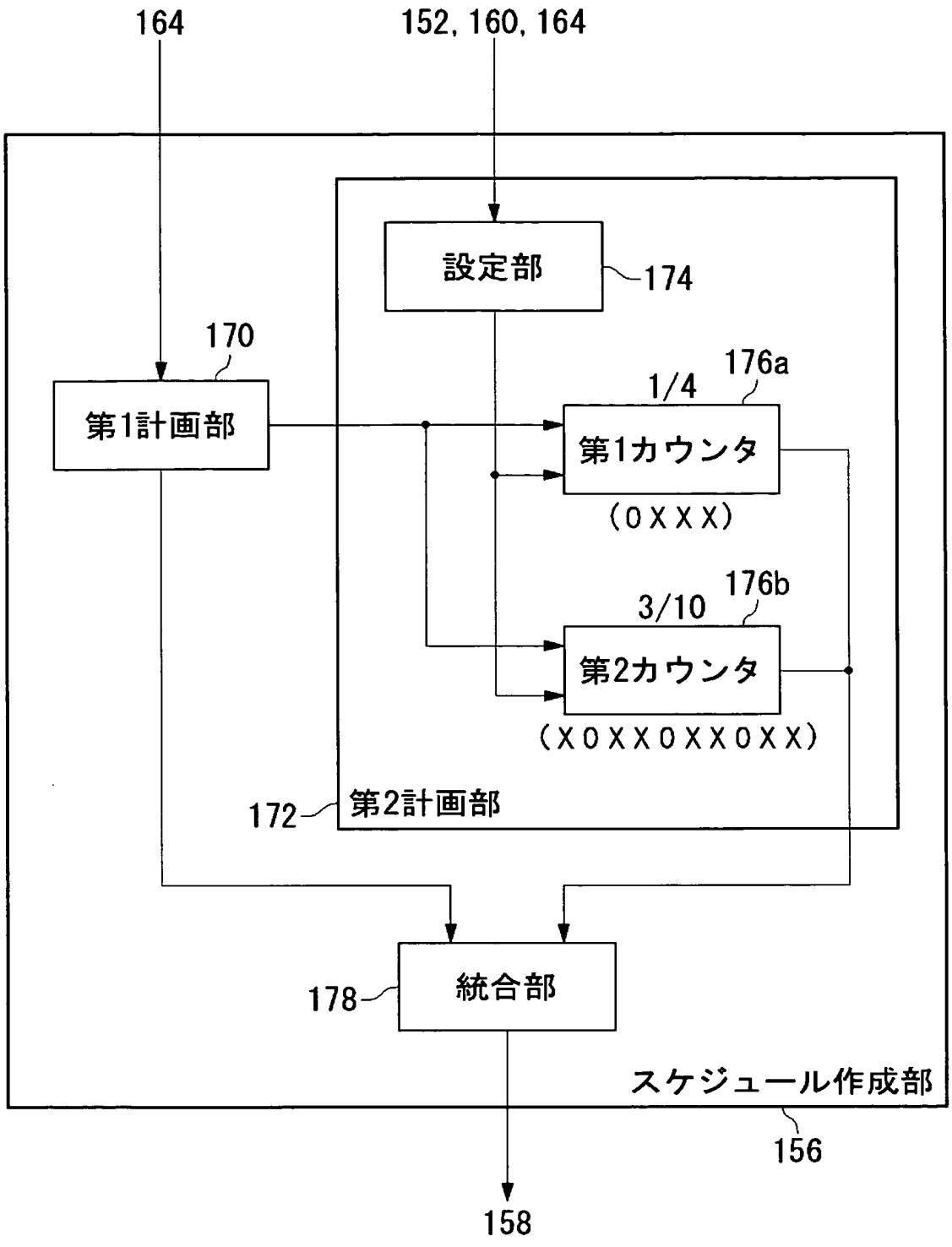


【図６】

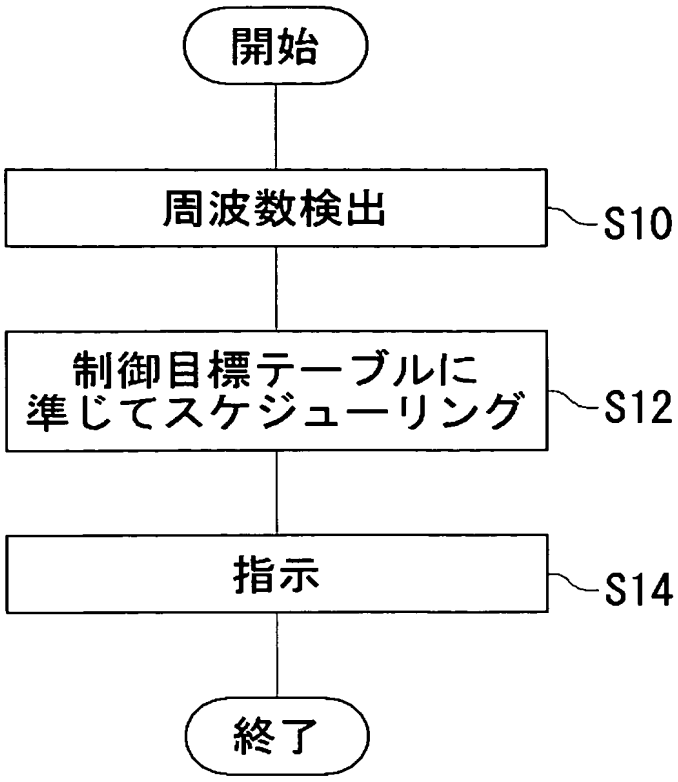


164

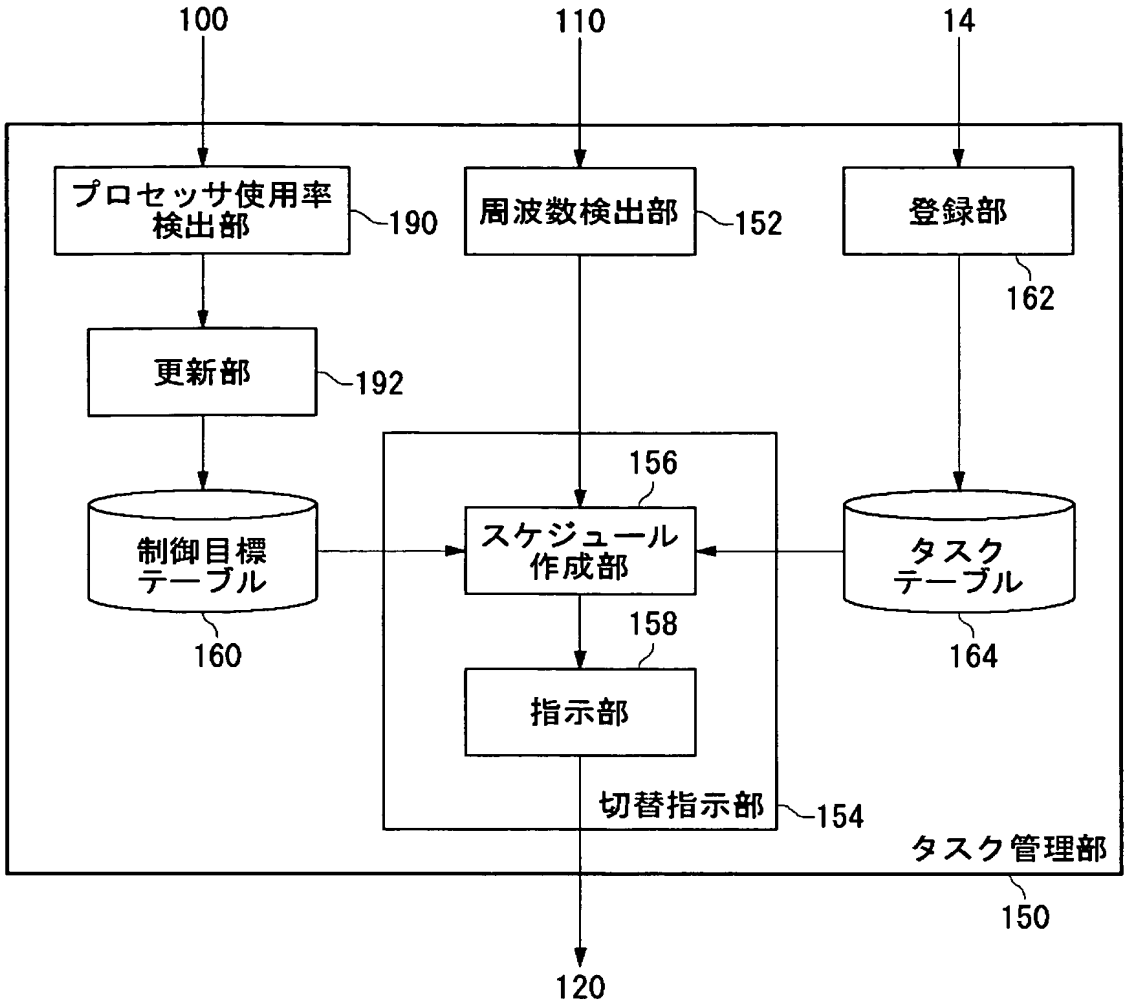
【図７】


164
152, 160, 164
設定部
174
170
第1計画部
1/4
176a
第1カウンタ
(0XXX)
3/10
176b
第2カウンタ
(X0XX0XX0XX)
172
第2計画部
178
統合部
スケジュール作成部
156
158

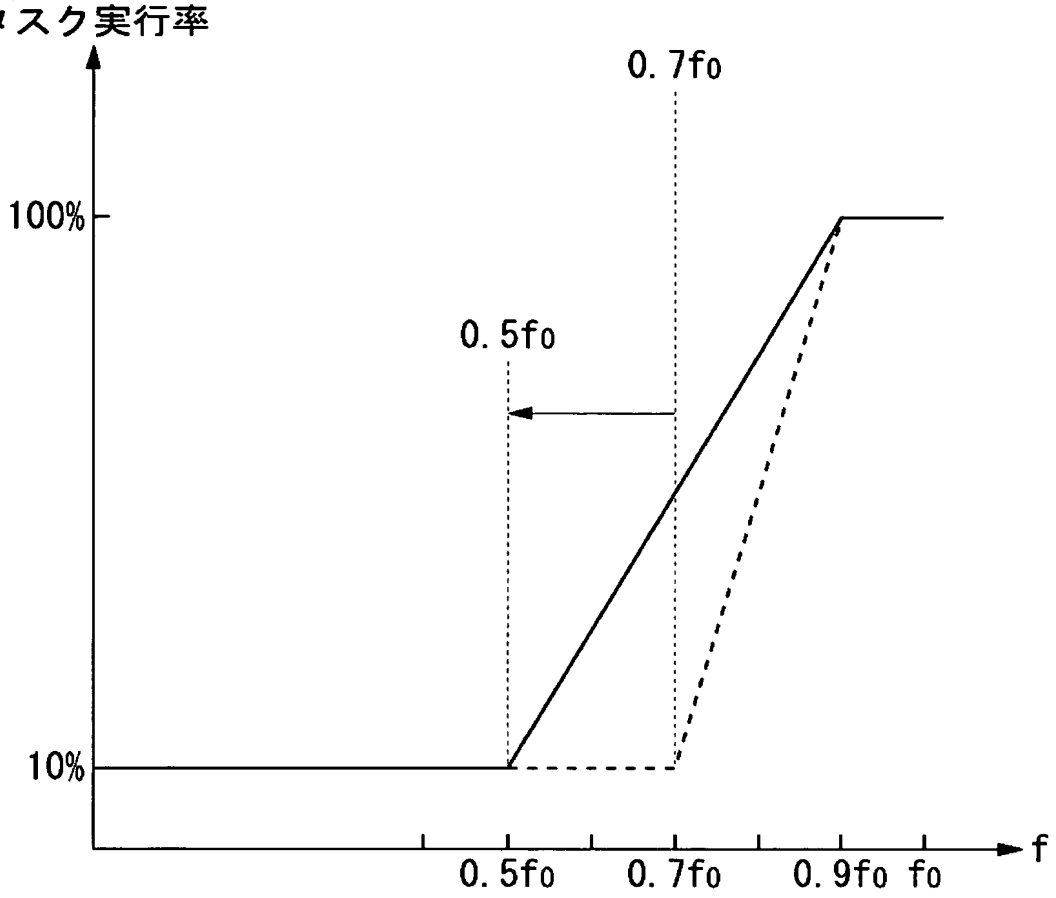
【図８】



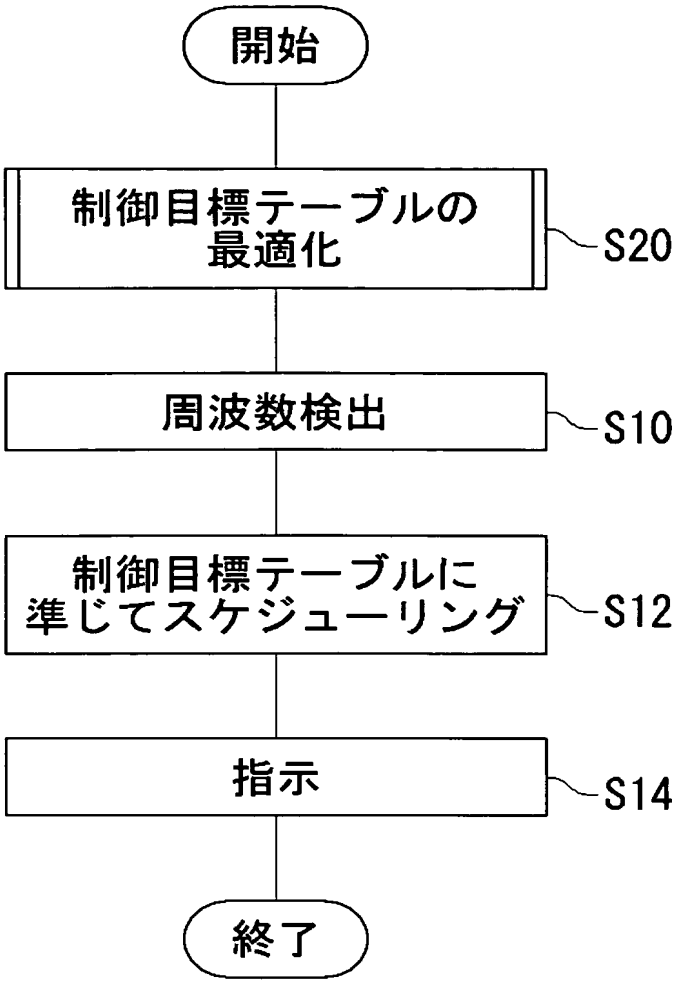
【図９】


100
110
14
プロセッサ使用率検出部
190
周波数検出部
152
登録部
162
更新部
192
156
制御目標テーブル
160
スケジュール作成部
タスクテーブル
164
158
指示部
切替指示部
154
タスク管理部
150
120

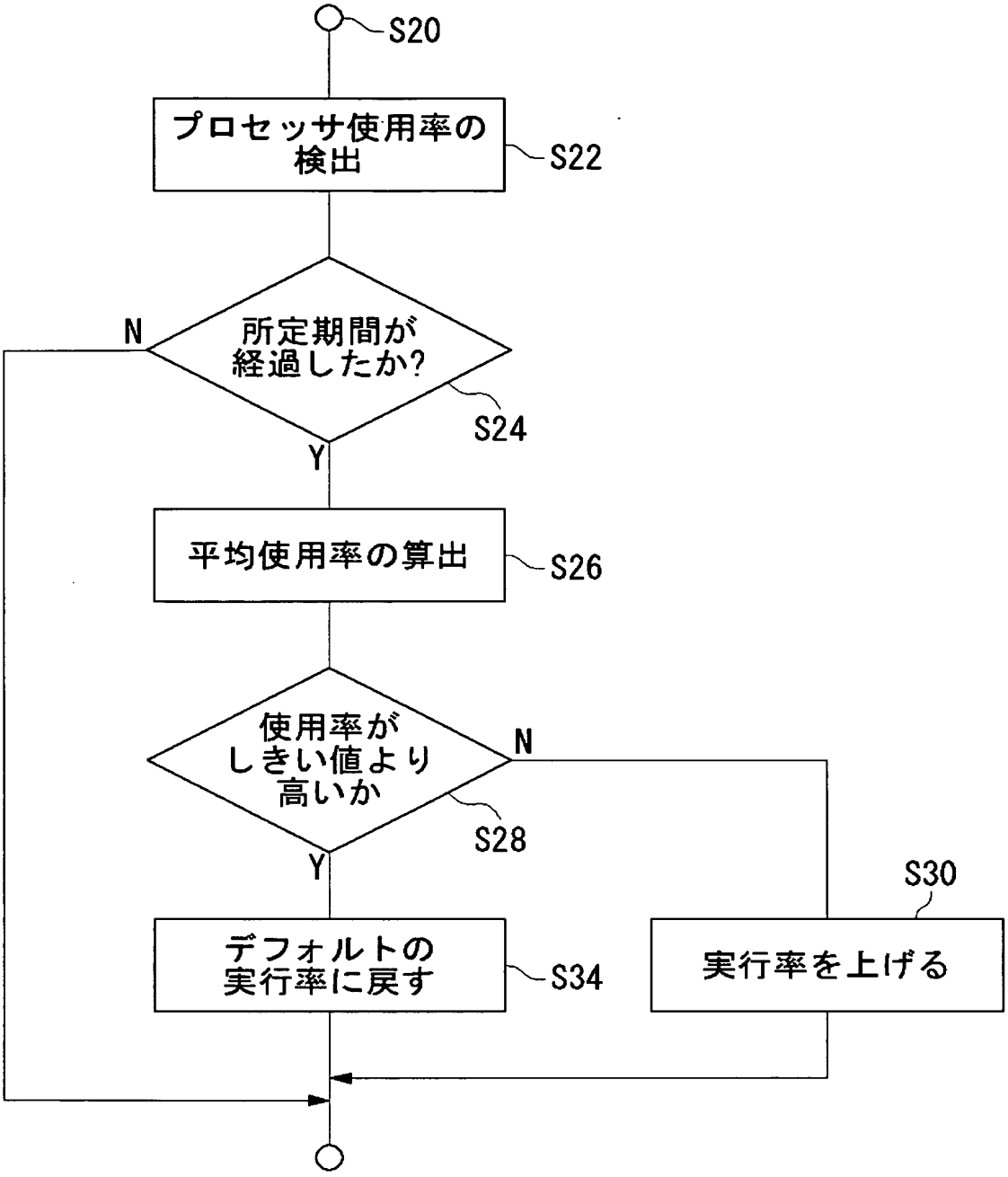
【図１０】



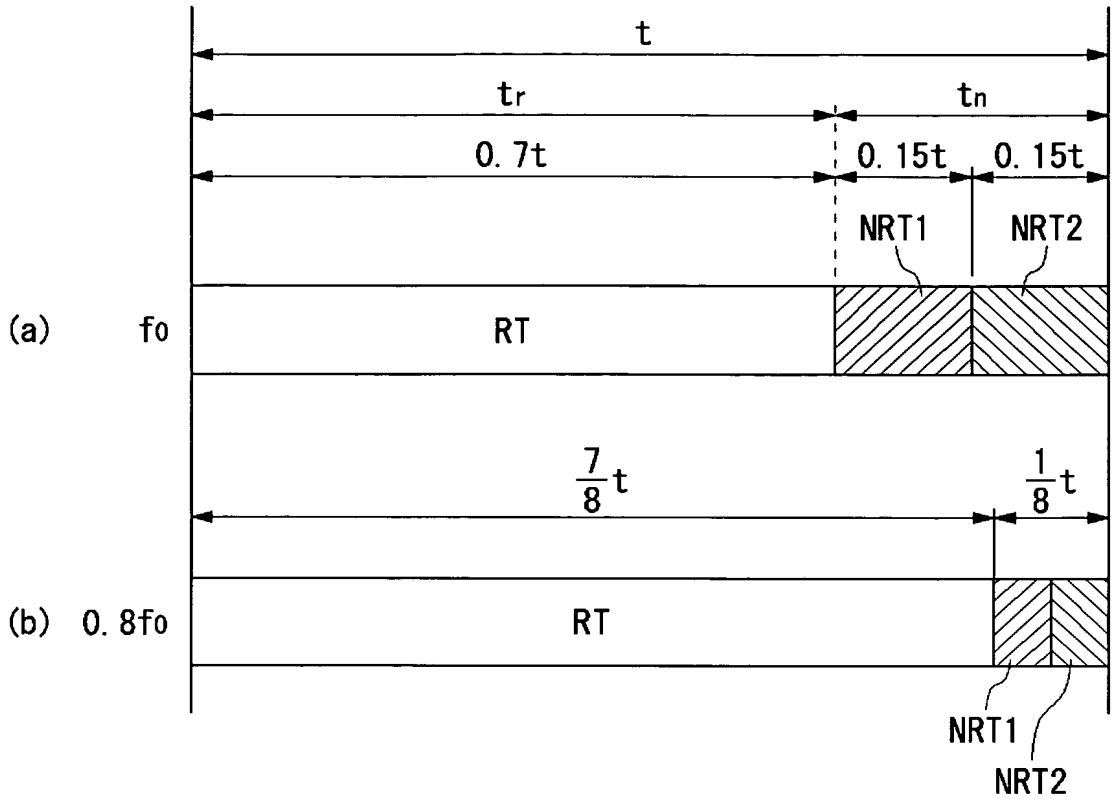
【図１１】

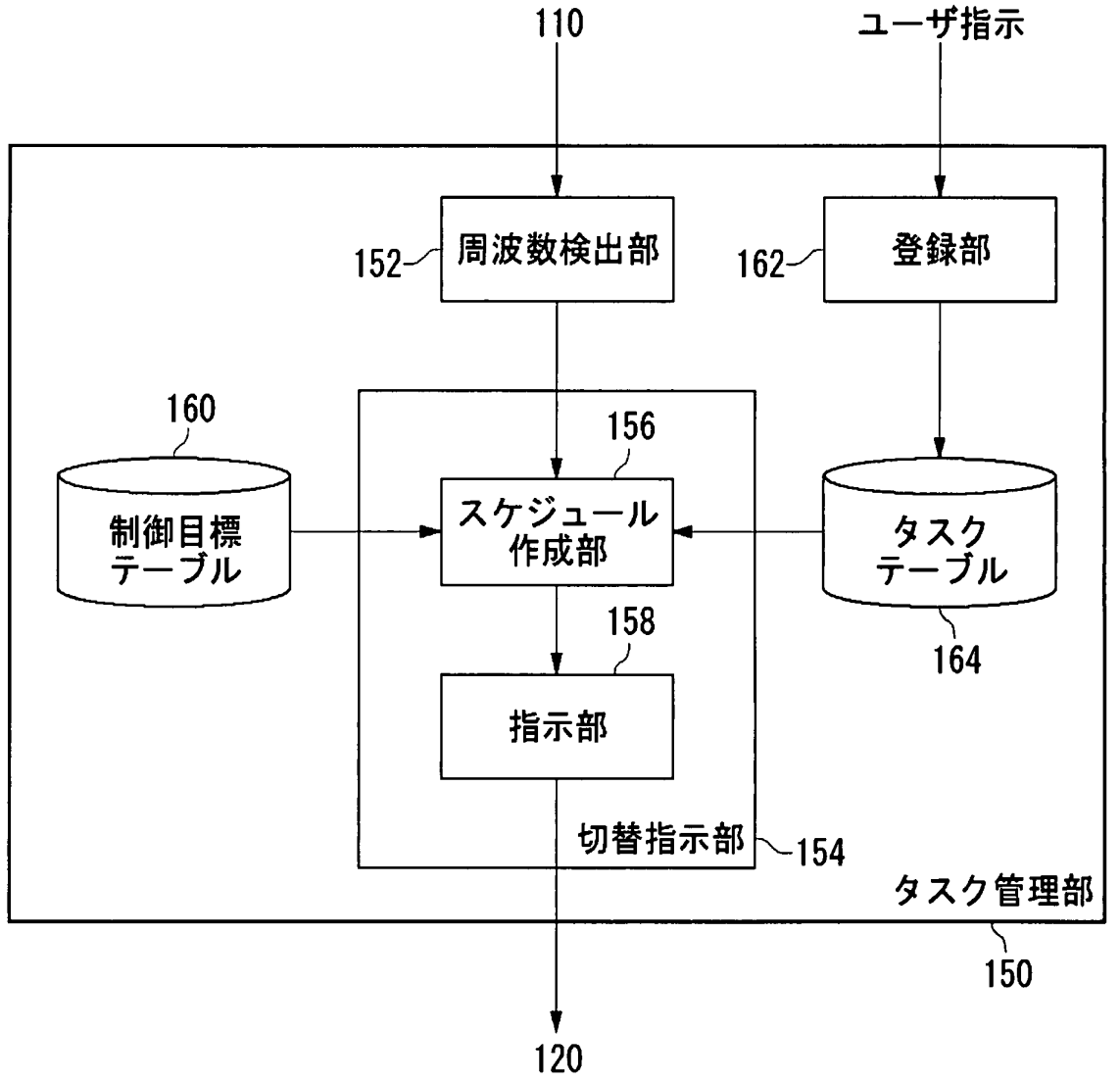

開始
制御目標テーブルの最適化
S20
周波数検出
S10
制御目標テーブルに準じてスケジューリング
S12
指示
S14
終了

【図１２】


S20
プロセッサ使用率の検出
S22
所定期間が経過したか?
S24
N
Y
平均使用率の算出
S26
使用率がしきい値より高いか
S28
N
Y
S30
デフォルトの実行率に戻す
S34
実行率を上げる

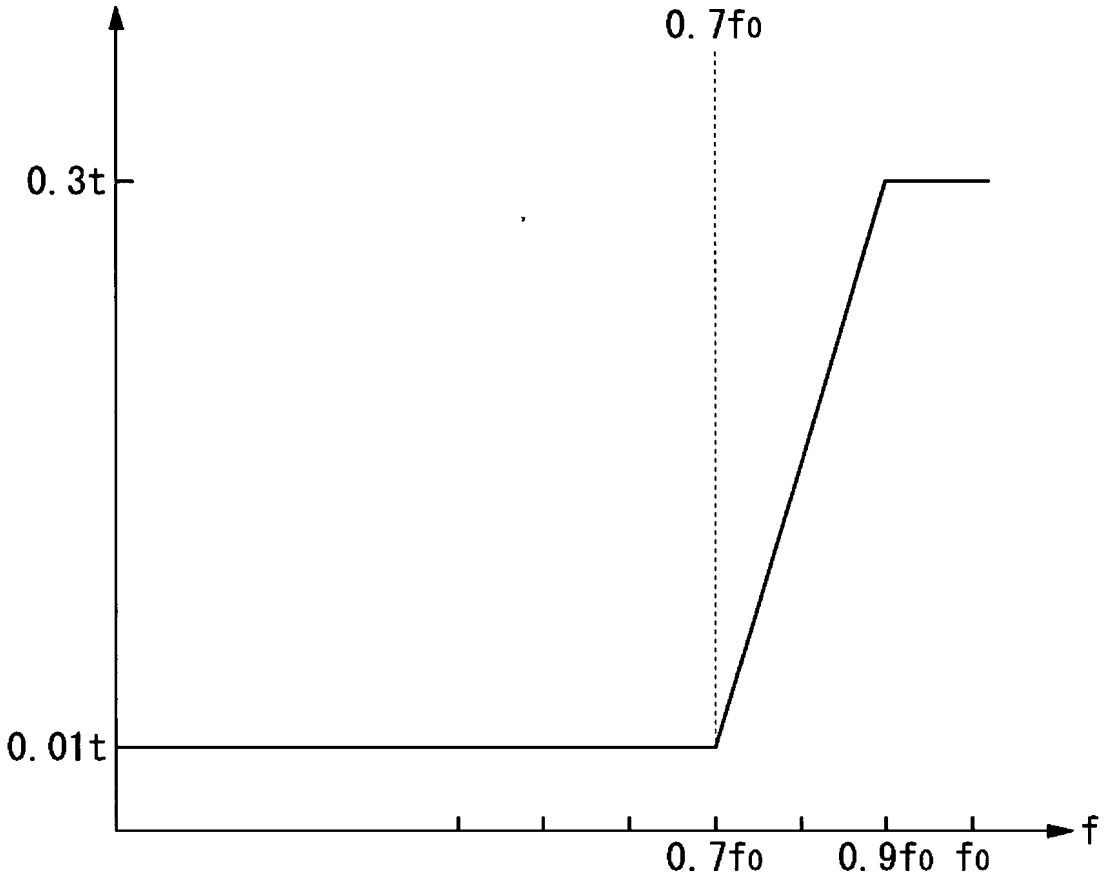
【図１３】


t
tr
tn
0.7t
0.15t
0.15t
NRT1
NRT2
(a)
fo
RT
7/8t
1/8t
(b)
0.8fo
RT
NRT1
NRT2

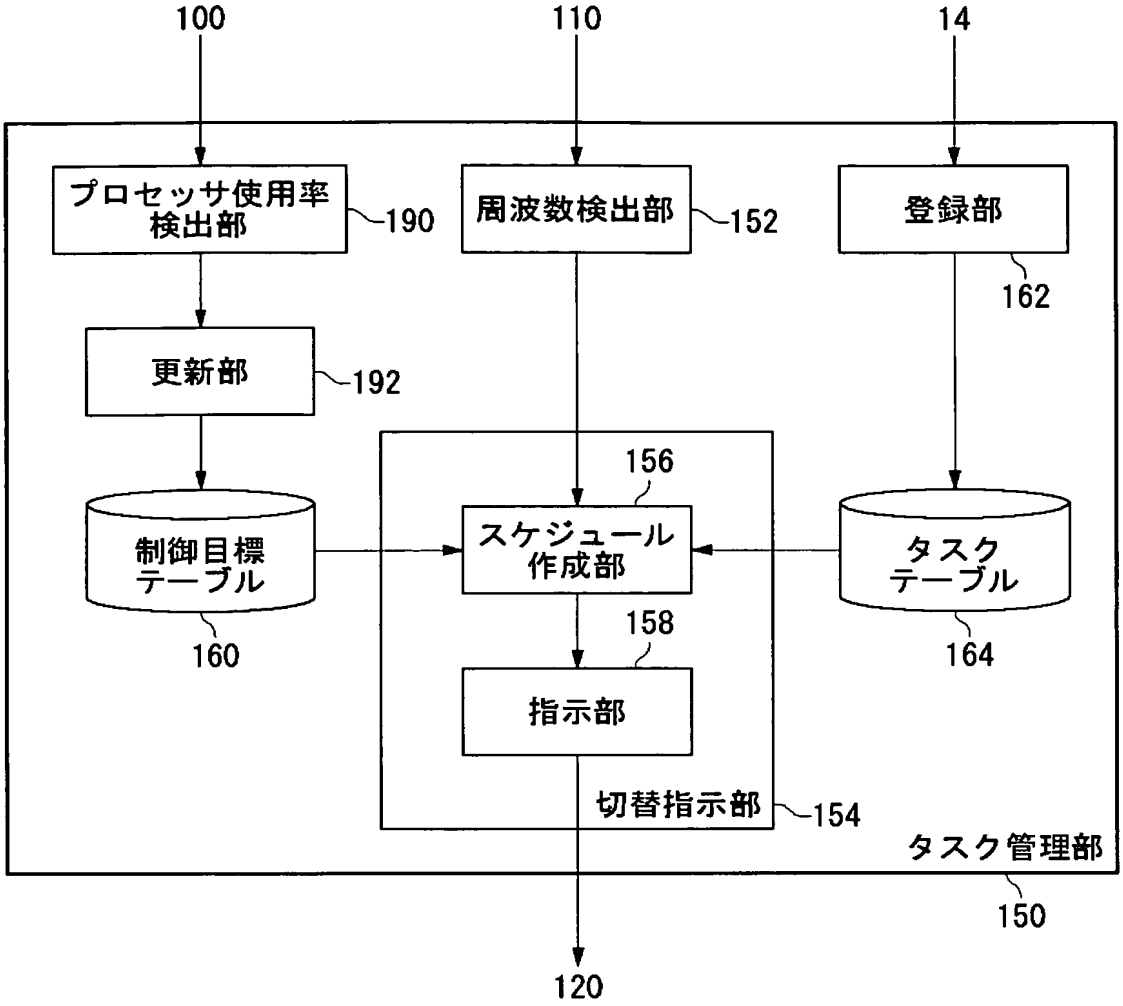

110
ユーザ指示
152
周波数検出部
162
登録部
160
156
制御目標
テーブル
スケジュール
作成部
タスク
テーブル
158
164
指示部
切替指示部
154
タスク管理部
150
120

【図１３】

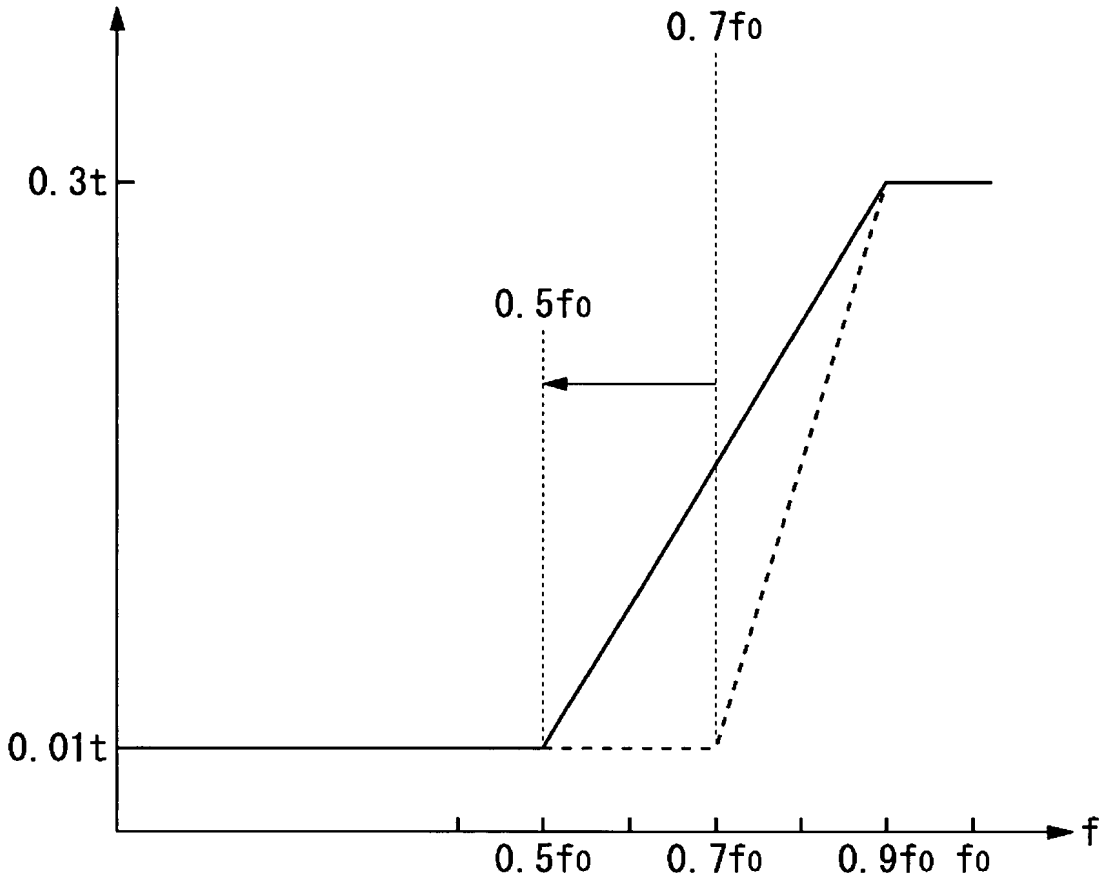
非保護期間



【図１６】


100
110
14
プロセッサ使用率検出部
190
周波数検出部
152
登録部
162
更新部
192
156
制御目標テーブル
160
スケジュール作成部
158
タスクテーブル
164
指示部
切替指示部
154
タスク管理部
150
120

【図１７】



【書類名】要約書

【要約】

【課題】　動作周波数を調整可能な半導体集積回路におけるタスクの管理が適切に行われていない。

【解決手段】　このタスク管理方法は、処理の単位時間をリアルタイム性を保証する保護帯域とリアルタイム性を保証しない非保護帯域に分割し、プロセッサの処理能力が低下したとき、非保護領域において実行すべきタスクの実行を適宜スキップするものである。すなわち、プロセッサの発熱を抑える目的で動作周波数を低くした場合、非保護帯域において実行すべきタスクをベストエフォートで処理する代わりに、保護帯域で実行すべきタスクのリアルタイム性を保証するものである。これにより、動作周波数が変動する場合にも適切にタスクの管理が行われ、プロセッサの処理能力が十分に発揮される。

【選択図】　図２

【書類名】　手続補正書
【整理番号】　SCEI03033
【提出日】　平成17年 1月17日
【あて先】　特許庁長官殿
【事件の表示】
　【出願番号】　特願2004-163649
【補正をする者】
　【識別番号】　395015319
　【氏名又は名称】　株式会社ソニー・コンピュータエンタテインメント
【代理人】
　【識別番号】　100105924
　【弁理士】
　【氏名又は名称】　森下　賢樹
　【電話番号】　03-3461-3687
【手続補正1】
　【補正対象書類名】　特許願
　【補正対象項目名】　発明者
　【補正方法】　変更
　【補正の内容】
　　【発明者】
　　【住所又は居所】　東京都港区南青山２丁目６番２１号　株式会社ソニー・コンピュータエンタテインメント内
　　【氏名】　安達　健一
　　【発明者】
　　【住所又は居所】　東京都港区南青山２丁目６番２１号　株式会社ソニー・コンピュータエンタテインメント内
　　【氏名】　矢澤　和明
　　【発明者】
　　【住所又は居所】　東京都港区南青山２丁目６番２１号　株式会社ソニー・コンピュータエンタテインメント内
　　【氏名】　瀧口　巖
　　【発明者】
　　【住所又は居所】　東京都港区南青山２丁目６番２１号　株式会社ソニー・コンピュータエンタテインメント内
　　【氏名】　今井　敦彦
　　【発明者】
　　【住所又は居所】　東京都港区南青山２丁目６番２１号　株式会社ソニー・コンピュータエンタテインメント内
　　【氏名】　田村　哲司
【その他】　出願の際、発明者中「瀧口　巖」の氏名を「瀧口　巌」と表記した誤記を訂正いたします。

３９５０１５３１９
20030701
住所変更

東京都港区南青山二丁目６番２１号
株式会社ソニー・コンピュータエンタテインメント

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/006966

International filing date: 08 April 2005 (08.04.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-163649
Filing date: 01 June 2004 (01.06.2004)

Date of receipt at the International Bureau: 26 May 2005 (26.05.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse